KU LEUVEN

ARENBERG DOCTORAL SCHOOL
Faculty of Engineering Science

On Kernels, Duality and Optimal Transport

Henri De Plaen

Supervisor:
Prof. dr. ir. Johan A. K. Suykens

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Electrical Engineering

July 2024

On Kernels, Duality and Optimal Transport

Henri DE PLAEN

Examination committee:
Prof. dr. ir. The Chairman, chair
Prof. dr. ir. Johan A. K. Suykens, supervisor
Prof. dr. ir. Panagiotis Patrinos
Prof. dr. ir. Karl Meerbergen
Prof. dr. Tinne Tuytelaars
Prof. dr. M. Cuturi
(Apple, ENSAE Paris)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Electrical Engineering

July 2024

Preface

...

Abstract

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Beknopte samenvatting

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

List of Abbreviations

MSE Mean Square Error. 2

OT Optimal Transport. 20, 22, 23

Unbalanced OT Unbalanced Optimal Transport. 20, 23

List of Symbols

$\cdot^\top$	Transpose operator
$\boldsymbol{I}_N$	Identity matrix of size $N \times N$.
$\boldsymbol{M}$	A matrix, assumed to lie in a space $\mathbb{R}^{N \times M}$.
$\boldsymbol{v}$	A vector, assumed to lie in a space $\mathbb{R}^N$.
a	A scalar, assumed to lie in $\mathbb{R}$.
N	A number, assumed to lie in $\mathbb{N}$.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

La mathématique est l'art de donner
le même nom à des choses différentes.

— Henri Poincaré

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie

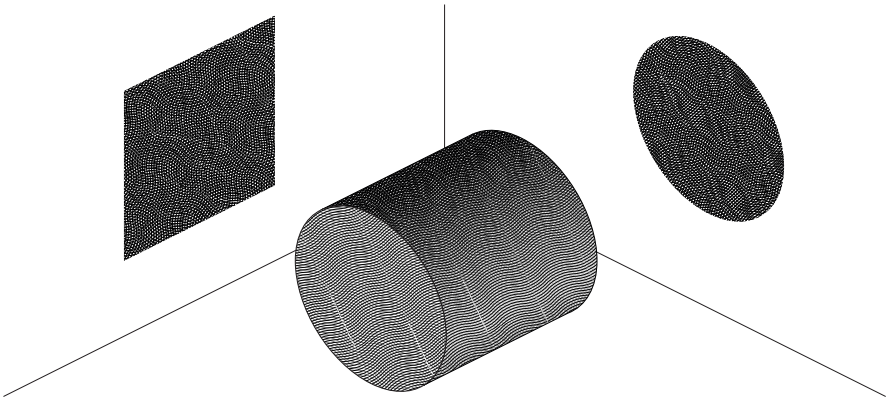


FIGURE 1.1: A same object can have different representations. Depending on the projection, a cylinder can be viewed either as a square or as a circle.

vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa. MSE

Textwidth: 341.43306pt

CHAPTER 2

Kernels

CHAPTER 3

Optimal Transport

CHAPTER 4

Wasserstein Exponential Kernels

In the context of kernel methods, the similarity between data points is encoded by the kernel function which is often defined thanks to the Euclidean distance; the squared exponential kernel is a common example. Recently, other distances relying on optimal transport theory – such as the Wasserstein distance between probability distributions – have shown their practical relevance for different machine learning techniques. In this paper, we study the use of exponential kernels defined thanks to the regularized Wasserstein distance and discuss their positive definiteness. More specifically, we define Wasserstein feature maps and illustrate their interest for supervised learning problems involving shapes and images. Empirically, Wasserstein squared exponential kernels are shown to yield smaller classification errors on small training sets of shapes, compared to analogous classifiers using Euclidean distances.

4.1 Introduction

Contemporary machine learning methods frequently rely on neural networks and shape recognition relies more specifically on convolutional neural networks. The big advantage of the latter is its ability to take into account the underlying structure of the data by treating neighboring pixels together. If these methods are often impressive by their performance, they are also known for their drawbacks such as a weak robustness and a difficult explainability. On the other side, although not always as accurate as neural networks, kernel methods are praised for their easy explainability and robustness. Another advantage of kernel methods is their versatility as they can easily be used in supervised and unsupervised methods, as well as

for generation [PSS]. In this paper, we emphasize the interest of choosing a particular kernel based on Wasserstein distance for classifying small datasets consisting of shapes.

In the context of kernel methods, squared exponential kernel functions are widely used, mainly because of their universal approximation properties and their empirical success. These Gaussians consist of the exponential of the negative Euclidean distance squared. However, the Euclidean distance might not always be appropriate to compare data points when datKHSas some particular structure. Indeed, it measures the correspondence of each feature independently of the other features. For example, let's consider the case of two identical 2D-shapes. When the two shapes overlap, their Euclidean distance is zero. However, if they do not overlap, their relative Euclidean distance becomes large although the shapes are identical. In other words, the Euclidean distance only compares each pixel at the same place on the grid and does not take the neighbouring pixels into account. The general structure of the features is not taken into account, only their strict correspondence. Another distance – the Wasserstein distance – gained popularity in recent years since it can incorporate the structure of the data if the dataset can be processed in such a manner that the datapoints can be considered as probability distributions.

Contributions

The contributions of this paper are the following. Empirically, we demonstrate that squared exponential kernels (4.1) based on a regularized Wassertein distance are performant on small scale classification problems involving shape datasets, compared for instance to the popular Gaussian RBF kernel [RW06]. Also, an approximation technique is proposed, with the so-called Wasserstein feature map, so that a positive semi-definite (psd) kernel can be defined from the Wasserstein squared exponential kernel which is not necessarily psd.

Notations and conventions

In the sequel, we denote vectors by bold lower case letters. Let $\mathbf{1}$ be the all ones column vector. Also, we define δ_y to be the Dirac measure at point y. A kernel $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is called positive semi-definite if all kernel matrices $K = [k(\boldsymbol{x_i}, \boldsymbol{x_j})]_{i,j=1}^n$ are positive semi-definite.

Wasserstein distances

The Wasserstein distance is a central notion in optimal transport theory. Also known as the *earth mover's distance*, it corresponds to the optimal transportation cost between two

measures [Vil08; PC19]. Let $p > 0$. We then define two normalized empirical measures $\boldsymbol{\alpha} = \sum_{i=1}^{m} a_i \delta_{\boldsymbol{y_i}}$ and $\boldsymbol{\beta} = \sum_{j=1}^{n} b_j \delta_{\boldsymbol{z_j}}$ such that $\boldsymbol{a}^\top 1 = 1$ and $\boldsymbol{b}^\top 1 = 1$, and where $\{\boldsymbol{y_i} \in \mathbb{R}^d\}_{i=1}^m$, $\{\boldsymbol{z_j} \in \mathbb{R}^d\}_{j=1}^n$ are support points. Also, we define a distance matrix $d_{ij} = d\left(\boldsymbol{y_i}, \boldsymbol{z_j}\right)$, *e.g.* the Euclidean distance $\|\boldsymbol{y_i} - \boldsymbol{z_j}\|_2$. Then, the p-Wasserstein distance is given by

$$\mathcal{W}_p(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \left(\min_{\boldsymbol{\pi} \in \Pi(\boldsymbol{\alpha}, \boldsymbol{\beta})} \sum_{i,j} \pi_{ij} d_{ij}^p \right)^{1/p},$$

with $\Pi(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \left\{ \Pi \in \mathbb{R}^{m \times n} | \Pi 1 = \boldsymbol{a} \text{ and } \Pi^\top 1 = \boldsymbol{b} \right\}$, the set of joint distributions π with specified marginals given by α and β. Intuitively, the optimal probability distribution $\pi^\star$ represents the optimal mass transportation scheme from α to β. A particular result occurs in the one-dimensional case ($d = 1$) assuming the support points are ordered, *i.e.*, $y_1 \leq ... \leq y_m$ and $z_1 \leq ... \leq z_n$ with $n = m$, where the Wasserstein distance reduces to an ℓ^p-norm: $\mathcal{W}_p^p \left(\frac{1}{n} \sum_{i=1}^n \delta_{y_i}, \frac{1}{n} \sum_{j=1}^n \delta_{z_j} \right) = \frac{1}{n} ||\boldsymbol{y} - \boldsymbol{z}||_p^p$ [PC19]. This connection between ℓ^p-norms and Wasserstein distances is only clear in one dimension, illustrating here again the fact that ℓ^p-norms do not take into account the underlying structure. To do so, we need to consider the case $d > 1$. In this way we can define the following kernel function

$$k_W(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \exp\left(-\frac{W_2^2(\boldsymbol{\alpha}, \boldsymbol{\beta})}{2\sigma^2} \right), \tag{4.1}$$

where $\sigma > 0$ is a bandwidth parameter.

This has however some undesirable consequences concerning positive definiteness. A kernel $k(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-t f(\boldsymbol{x}, \boldsymbol{y})\right)$ is positive semi-definite for all $t > 0$ if and only if $f(\boldsymbol{x}, \boldsymbol{y})$ is Hermitian and *conditionally* negative semi-definite [BCR84]. Recall that a kernel is *conditionally* negative semi-definite if any Gram matrix $F = [f(\boldsymbol{x_i}, \boldsymbol{x_j})]_{i,j=1}^n$ (with $n \geq 2$) built from a discrete sample satisfies $\boldsymbol{c}^\top F \boldsymbol{c} \leq 0$ for all $\boldsymbol{c}$ such that $1^\top \boldsymbol{c} = 0$. However, the Wasserstein distance for $d > 1$ is not necessarily *conditionally* negative definite [PC19]. By consequence, we cannot guarantee that any resulting squared exponential kernel matrix built with the 2-Wasserstein distance is positive definite. This property is fundamental in kernel theory and more specifically for defining *reproducing kernel Hilbert spaces* (RKHS; see [SS01a] for more details).

4.2 Dealing with indefinite exponential kernels

This restriction has lead authors to consider only some specific cases of Wasserstein distances which are known to be positive definite. The one-dimensional generic case is proven to be

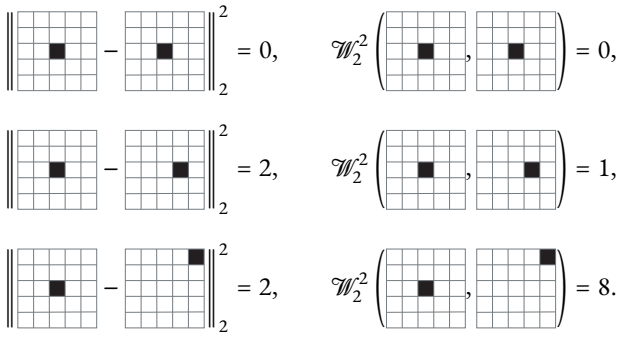


FIGURE 4.1: Due to the incorporation of the cost, the Wasserstein distance is able to capture the underlying structure of an image. The pixels are not considered to be all equal to each other as with Euclidean norms, but their relative distances is taken into account.

positive definite and has lead to the introduction of sliced Wasserstein distances [CCO17; KNS+19]. Another notable case is the Wasserstein distance between two multivariate normal distributions in more than one dimension, which can even be written in closed form [PC19].

Some kernel methods have been used with indefinite kernels, such as LS-SVMs [SGB+02; HMHS17]. This leads however to a slightly different interpretation of the global problem, using Kreĭn spaces for which a weaker version of the representer theorem holds [OMCS04]. In this paper, we propose an alternative which allows to continue working with a positive definite kernel approximating the squared exponential kernel. If the Wasserstein exponential kernel can not be used, we can always find a parameter $\sigma > 0$ and a finite dimensional feature map resulting in a positive definite kernel.

4.2.1 Positive definite squared exponential kernels and bandwidth choice

In this section, we show that for a given dataset, the corresponding Gram matrix of k_W is positive definite if the bandwidth parameter $\sigma > 0$ is small enough.

Definition 4.1. *Let $d : \mathcal{D} \times \mathcal{D} \to \mathbb{R}_{\geq 0}$ be a symmetric function such that $d(\boldsymbol{x}, \boldsymbol{x}) = 0$ and let $\{\boldsymbol{x}_i \in \mathcal{D}\}_{i=1}^{N}$ be a dataset. A squared exponential kernel matrix is defined as*

$$\boldsymbol{K}_{d,\sigma} = \left[\exp\left(\frac{-d^2(\boldsymbol{x}_i, \boldsymbol{x}_j)}{2\sigma^2}\right)\right]_{i,j=1}^{N}.$$

By construction, this squared exponential kernel matrix will be symmetric and have a diagonal consisting only of ones. Its eigenvalues are real. To investigate its (semi)-definiteness, we have to investigate the sign of the minimum eigenvalue. The minimum eigenvalue $\lambda_{\min}(\sigma)$ of $\boldsymbol{K}_{d,\sigma}$ is the function $\lambda_{\min} : \mathbb{R}_{>0} \to \mathbb{R}, \sigma \mapsto \min\{\lambda_1, \dots, \lambda_N\}$ where $\lambda_1, \dots, \lambda_N$ are the eigenvalues of $\boldsymbol{K}_{d,\sigma}$. We can now prove the following results.

Lemma 4.1. *The eigenvalues of the squared exponential kernel matrix $\boldsymbol{K}_{d,\sigma}$ are continuous functions of σ. In particular, $\lambda_{\min}(\sigma)$ is continuous.*

Proof. This is a direct consequence of the continuity of the roots of a polynomial with continuous coefficients. Therefore, we have to prove that the coefficients of the characteristic polynomial of the squared exponential kernel matrix $\boldsymbol{K}_{d,\sigma}$ is continuous as a function of σ. The characteristic polynomial is given by $\det(\boldsymbol{K}_{d,\sigma} - \lambda \boldsymbol{I})$ and by the formula of Leibniz, we ultimately have that the characteristic polynomial is a sum of products of elements of $\boldsymbol{K}_{d,\sigma} - \lambda \boldsymbol{I}$, which are continuous in function of σ. Hence, the coefficients are continuous and so are the eigenvalues. ■

Lemma 4.2. $\lim_{\sigma \to 0} \boldsymbol{K}_{d,\sigma} = \mathrm{id}$ *and thus* $\lambda_{\min}(0) = 1$.

Proof. From Definition 4.1, we know that $\left[\boldsymbol{K}_{d,\sigma}\right]_{i,j} = \exp\left(\frac{-d^2(\boldsymbol{x}_i, \boldsymbol{x}_j)}{2\sigma^2}\right)$ with $d^2(\boldsymbol{x}_i, \boldsymbol{x}_i) = 0$ and $d^2(\boldsymbol{x}_i, \boldsymbol{x}_j) > 0$ for $i \neq j$. Denote $C_{i,j} = d^2(\boldsymbol{x}_i, \boldsymbol{x}_j)$ for simplicity. We have $\lim_{\sigma \to 0} \exp\left(\frac{0}{2\sigma^2}\right) = 1$ and $\lim_{\sigma \to 0} \exp\left(-\frac{C_{i,j}}{2\sigma^2}\right) = 0$ with $C_{i,j} > 0$ for $i \neq j$, thus the identity matrix. By consequence, all the eigenvalues are equal to 1. ■

Lemma 4.3. *We have* $\lim_{\sigma \to \infty} \boldsymbol{K}_{d,\sigma} = 11^T$ *and thus* $\lim_{\sigma \to \infty} \lambda_{\min}(\sigma) = 0$.

Proof. Similarly, we have $\lim_{\sigma \to +\infty} \left[\boldsymbol{K}_{d,\sigma}\right]_{i,j} = 1$ everywhere. By consequence, we have $\lambda_{\max} = N$ and all others equal to zero, hence $\lambda_{\min} = 0$. ■

Proposition 4.1. *There exists a $\sigma_{\mathrm{PSD}} \in \mathbb{R}_+$ such that $\boldsymbol{K}_{d,\sigma}$ is positive semi-definite for all $\sigma \leq \sigma_{\mathrm{PSD}}$.*

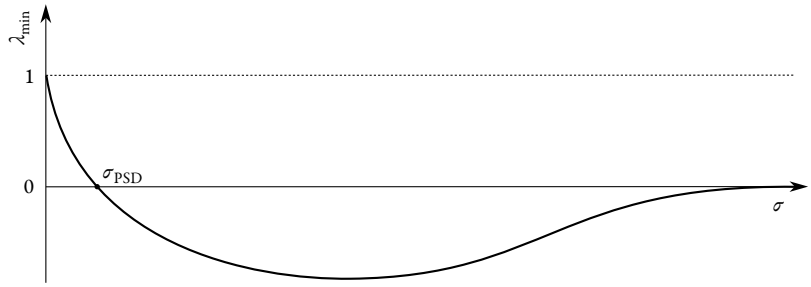


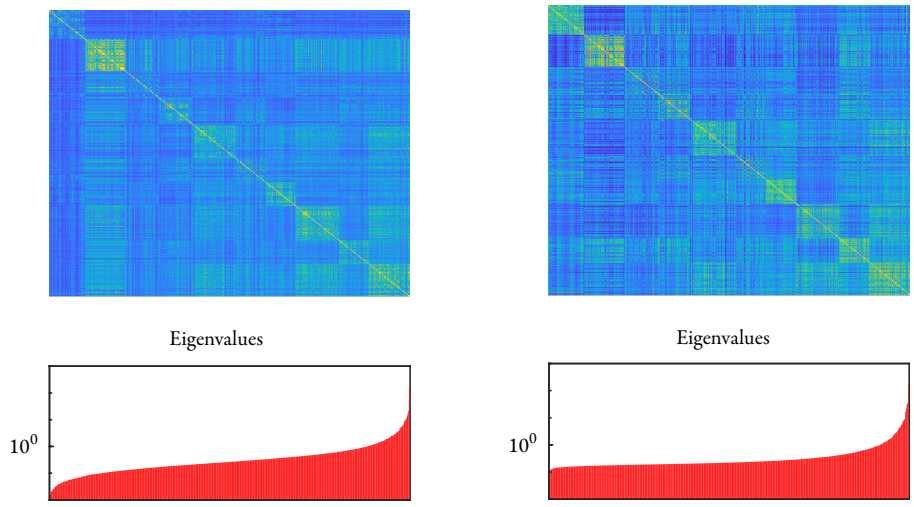
FIGURE 4.2

Proof. Let us proceed *ad absurdum* and suppose this is not the case. We consider the sequence $\left(\sigma_n\right)_n$ converging to 0 with $\sigma_0 = \sigma_{\text{PSD}}$. There must exist some subsequence $\left(\sigma_{n_j}\right)_j$ such that $\left(\lambda_{\min}\left(\sigma_{n_j}\right)\right)_j < 0$. If this sequence is finite, then it is sufficient to consider a new sequence with $\sigma_{\text{PSD}} = \sigma_{n_{j_{\max}+1}}$. If this subsequence is infinite, then $\left(\lambda_{\min}\left(\sigma_n\right)\right)_n$ cannot converge to 1. This is impossible because of the continuity of $\lambda_{\min}(\sigma)$ (Lemma 4.1) and its convergence to 1 (Lemma 4.2). Hence, there exist some $\sigma_{\text{PSD}} > 0$ such that $\lambda_{\min}(\sigma) \geq 0$ for all $\sigma \leq \sigma_{\text{PSD}}$. This proves our proposition. ■

We can empirically see the result of Proposition 4.1 in Fig. 4.3, where all eigenvalues are positive. Intuitively, decreasing the bandwidth σ tends to make the smallest distances more predominant, pushing the smallest eigenvalue progressively to the positive side. In this sense, an indefinite kernel matrix with σ close to σ_{PSD} will lead to proportionally very small negative eigenvalues in magnitude. In this case, a finite positive definite approximation can be justified.

4.2.2 Wasserstein features

We can consider a finite dimensional feature map $\boldsymbol{\phi}(\boldsymbol{x})$ such that the positive semi-definite kernel $\boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{\phi}(\boldsymbol{y})$ approximates $k_W(\boldsymbol{x}, \boldsymbol{y})$ given in (4.1). This finite approximation is based on a training dataset $\{\boldsymbol{x}_i\}_{i=1}^N$ for constructing an original kernel matrix $\boldsymbol{K} = \left[k_W(\boldsymbol{x}_i, \boldsymbol{x}_j)\right]_{i,j=1}^N \in \mathbb{R}^{N \times N}$. It suffices to truncate the spectral decomposition of the kernel matrix $\boldsymbol{K} = \sum_{l=1}^N \lambda_l \boldsymbol{v}_l \boldsymbol{v}_l^\top$ to the ℓ largest strictly positive eigenvalues. This will result in a new positive semi-definite kernel matrix $\boldsymbol{K}^{(\ell)} \overset{\text{def}}{=} \sum_{l=1}^{\ell} \lambda_l \boldsymbol{v}_l \boldsymbol{v}_l^\top \succeq 0$ with $\lambda_1 \geq \cdots \geq \lambda_N$. We can now reconstruct the



(a) Euclidean (b) Wasserstein

FIGURE 4.3: Comparison of the classical squared exponential kernel matrix (based on a ℓ^2-distance) and the introduced Wasserstein exponential kernel matrix on 500 normalized digits of the MNIST dataset [LC10]. The digits are ordered by class in ascending order. For the color legend, please refer to Fig. 4.4.

different components of an approximate feature map

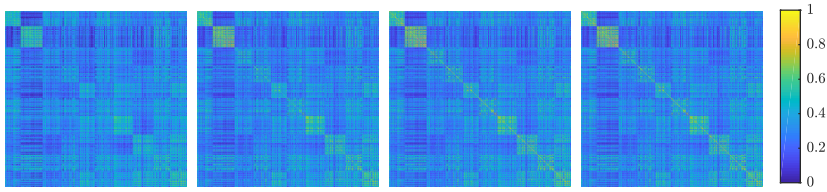
$$\phi_l(\boldsymbol{x}) \stackrel{\text{def}}{=} \frac{1}{\sqrt{\lambda_l}} \boldsymbol{k}_{\boldsymbol{x}}^\top \boldsymbol{v}_l, \qquad \text{for } i = 1, \dots, \ell, \tag{4.2}$$

with $\boldsymbol{k}_{\boldsymbol{x}} \stackrel{\text{def}}{=} \begin{bmatrix} k_W(\boldsymbol{x}, \boldsymbol{x}_1) & \cdots & k_W(\boldsymbol{x}, \boldsymbol{x}_N) \end{bmatrix}^\top$. We refer to these different components as the *Wasserstein features* as they compose the approximate feature map $\boldsymbol{\phi}(\boldsymbol{x}) \stackrel{\text{def}}{=} \begin{bmatrix} \phi_1(\boldsymbol{x}) & \cdots & \phi_\ell(\boldsymbol{x}) \end{bmatrix}^\top$ of the Wasserstein exponential kernel. This approximate feature map is constructed by using a training dataset, but can afterwards be evaluated at any out-of-sample point. By construction, we can verify that the Wassertein features evaluated on the training dataset result in the truncated kernel matrix.

Proposition 4.2. *We have* $\left[\boldsymbol{\phi}(\boldsymbol{x}_i)^\top \boldsymbol{\phi}(\boldsymbol{x}_j)\right]_{i,j=1}^N = \boldsymbol{K}^{(\ell)}$.

Proof. It suffices to observe that $\boldsymbol{k}_{\boldsymbol{x}_i} = \sum_{l=1}^N \lambda_l \boldsymbol{v}_l \left[\boldsymbol{v}_l\right]_i$. By consequence, we have $\phi_l(\boldsymbol{x}_i) = \sqrt{\lambda_l} \left[\boldsymbol{v}_l\right]_i$. ■

Proposition 4.1 suggests that even if no suitable σ can be found such that the kernel matrix is psd, the negative eigenvalues will remain very small in magnitude. By consequence, we can suppress them without much information loss. A truncated kernel is thus very close to the original one in spectral norm. This justifies the Wasserstein features in this sense that they are very close to the Wasserstein exponential kernel as well as being positive definite by construction. This fact can be visualized on Fig. 4.4.



(a) $\ell = 5$ (b) $\ell = 15$ (c) $\ell = 500$ (d) Exact

FIGURE 4.4: Kernel matrices constructed as the inner products of a different number Wasserstein features of a test set. These matrices are compared with the exact Wasserstein squared exponential kernel matrix of the test set. Both the training set and the test set are of size $N = 500$.

Clearly, the *Wasserstein features* yield a positive semi-definite kernel. Moreover, it is also advantageous to work with finite dimensional feature maps to reduce the training time. Indeed, the computation of the Wasserstein distance (or an approximation with *e.g.* Sinkhorn's algorithm [Cut13a]) is still relatively expensive compared to ℓ^2 distance.

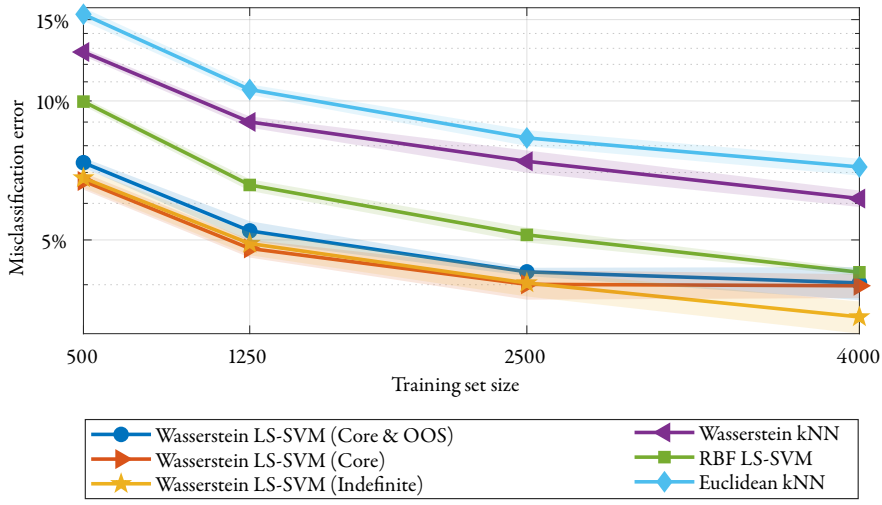
4.3 Experiments

4.3.1 Setup for 2D shape classification

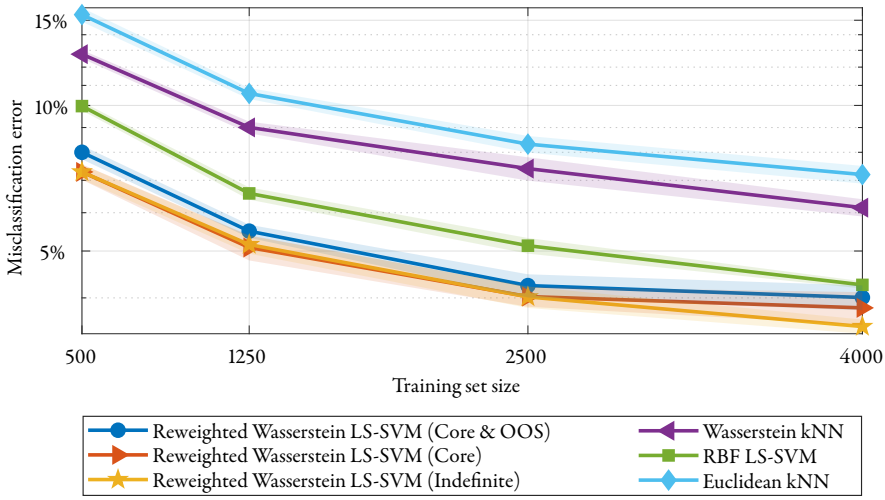
Let $\boldsymbol{u}$ be a greyscale image that we unfold as a vector of length m and so that $u_i > 0$ is the "grey" value at the pixel $\boldsymbol{y_i}$ of a pixel grid. It is mapped to a probability $\boldsymbol{\alpha} = \sum_{i=1}^{m} a_i \delta_{\boldsymbol{y_i}}$ by defining $a_i = u_i / \|\boldsymbol{u}\|_1$, so that the mass of $\boldsymbol{\alpha}$ is one. In practice, the $p = 2$ Wassertein distance is computed in this paper with the help of the well-known entropic regularization

$$\mathcal{W}_{2,\epsilon}^2(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \min_{\boldsymbol{\pi} \in \Pi(\boldsymbol{\alpha}, \boldsymbol{\beta})} \sum_{i,j} \left(\pi_{ij} d_{ij}^2 + \epsilon \pi_{ij} \log \pi_{ij} \right),$$

where $\epsilon > 0$ is a small regularization term and d_{ij}^2 is the Euclidean distance between pixels located at $\boldsymbol{y_i}$ and $\boldsymbol{y_j}$ in a pixel grid. The advantage of this regularized problem is that its solution



(a) Comparison of Wasserstein exponential kernels with other similar methods.



(b) Comparison of *reweighted* Wasserstein exponential kernels with other similar methods.

Figure 4.5: Mean misclassification rates for various subset sizes of the MNIST dataset, computed on 7 simulations. The standard deviation is given by the errors bars. For the specific case of "Core + OOS", the out-of-sample subset represents 300 datapoints on 500, 750 on 1250, 1500 on 2500 and 2500 on 4000. The size of validation set is always 5000 and of the test set always 10 000.

TABLE 4.1: Percentage of classification error on the test set of three datasets. The standard deviation is given in parenthesis. The number of repeated simulations is 7 for MNIST, 8 for Quickdraw and 6 for USPS.

Dataset	**MNIST**		**Quickdraw**		**USPS**	
Method	**Avg.**	**Best**	**Avg.**	**Best**	**Avg.**	**Best**
Wass. LS-SVM (Core+OOS)	3.95 (± 0.18)	3.74	11.45 (± 0.39)	10.97	6.77 (± 0.52)	6.20
Wass. LS-SVM (Core)	3.81 (± 0.34)	3.28	10.80 (± 0.19)	10.52	7.93 (± 1.45)	6.35
Wass. LS-SVM (Indef.)	**3.40** (± 0.11)	**3.23**	**10.75** (± 0.27)	10.35	6.15 (± 0.67)	5.45
R. Wass. LS-SVM (Core+OOS)	3.91 (± 0.27)	3.45	11.79 (± 0.48)	10.95	6.68 (± 0.80)	5.70
R. Wass. LS-SVM (Core)	3.71 (± 0.15)	3.46	10.99 (± 0.44)	**10.07**	6.35 (± 0.11)	6.20
R. Wass. LS-SVM (Indef.)	3.48 (± 0.13)	3.29	12.43 (± 0.43)	11.95	**5.70** (± 0.29)	**5.40**
Wass. kNN	6.31 (± 0.33)	5.81	12.26 (± 0.33)	11.91	6.60 (± 0.44)	6.00
RBF LS-SVM	4.26 (± 0.10)	4.07	11.46 (± 0.20)	11.23	6.75 (± 0.04)	6.70
ℓ^2 kNN	7.20 (± 0.15)	6.95	15.32 (± 0.40)	14.68	7.52 (± 0.38)	7.20
Set size	**Core + OOS**	**Others**	**Core + OOS**	**Others**	**Core + OOS**	**Others**
Training	1500 + 2500	4000	500 + 750	1250	1000 + 1500	2500
Validation	5000	5000	5000	5000	2000	2000
Test	10 000	10 000	10 000	10 000	2000	2000

can be efficiently obtained thanks to the Sinkhorn algorithm, which can be parallelized. For more details, we refer to [PC19]. All the simulations used $\epsilon = 2.5$ and the diagonal of the distance matrix set to zero. This choice of value was motivated by trial-and-error in order to make the regularization parameter large enough to be close to the exact Wasserstein distance, without being too large, which increases drastically the computation time and eventually leads to unstable iterations of Sinkhorn's algorithm.

The full Wasserstein kernel matrix has N^2 elements, with N the number of datapoints. Computing the full kernel matrix requires thus to compute $N^2/2$ pairwise distances as they are symmetric. By using the Wasserstein features, the number of pairwise distances to compute can be reduced to $N_1^2/2 + N_1 N_2$, where N_1 is the size of the training dataset and N_2 of the out-of-sample dataset. The computation of each pairwise Wasserstein distance has a complexity of $\widetilde{O}\left(n^2/\epsilon^3\right)$ where n is the dimension of each datapoint and ϵ the regularization parameter [ANR17a], compared to $O(n)$ for the Euclidean distance. However, Sinkhorn's algorithm can be implemented on GPUs, benefiting from their massive parallelization capabilities, allowing to compute pairwise distances from an arbitrary group of datapoints to a common datapoint together.

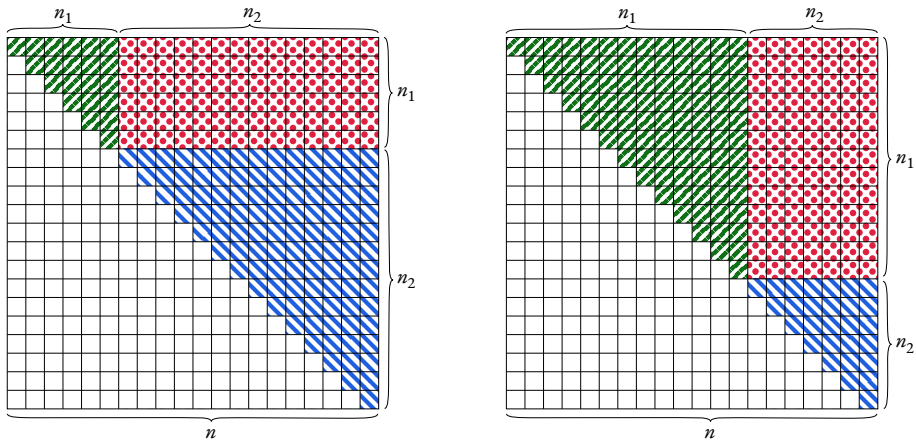


FIGURE 4.6

4.3.2 Shape recognition

We illustrate the use of the Wasserstein based kernels in the context of shape classifications. Namely, we train a Least Squares Support Vector Machine [SV99a] classifier on subsets of the MNIST [LC10], Quickdraw[1] and USPS [Hul94] datasets, which are sampled uniformly at random. These three datasets contain handwritten digits and shapes. The multiclass problem is solved by a one-versus-one encoding. One instance of these binary classifiers $f(\boldsymbol{x}) = \operatorname{sign}(\boldsymbol{w}^{\star\top}\boldsymbol{\phi}(\boldsymbol{x}) + b^{\star})$ is obtained by solving

$$\min_{\substack{\boldsymbol{w}\in\mathbb{R}^{\ell};b\in\mathbb{R}\\ e_i\in\mathbb{R}}} \boldsymbol{w}^{\top}\boldsymbol{w} + \frac{\gamma}{N}\sum_{i=1}^{N} e_i^2 \text{ s.t. } e_i = y_i - \boldsymbol{w}^{\top}\boldsymbol{\phi}(\boldsymbol{x}_i) - b, \tag{4.3}$$

where $y_i \in \{-1, 1\}$ and $\boldsymbol{\phi}(\boldsymbol{x}) \in \mathbb{R}^{\ell}$ is a feature map obtained for instance thanks to (4.2). The solution is obtained by solving

$$\begin{bmatrix} \sum_i \boldsymbol{\phi}(\boldsymbol{x}_i)\boldsymbol{\phi}(\boldsymbol{x}_i)^{\top} + \frac{N}{\gamma}\mathbb{I} & \sum_i \boldsymbol{\phi}(\boldsymbol{x}_i) \\ \sum_i \boldsymbol{\phi}(\boldsymbol{x}_i)^{\top} & N \end{bmatrix} \begin{bmatrix} \boldsymbol{w} \\ b \end{bmatrix} = \begin{bmatrix} \sum_i y_i\boldsymbol{\phi}(\boldsymbol{x}_i) \\ \sum_i y_i \end{bmatrix},$$

which is a $(\ell + 1) \times (\ell + 1)$ linear system. A classifier can also be obtained by solving the dual problem of (4.3). The optimality conditions of this dual problem yield the following

[1] `https://quickdraw.withgoogle.com/data`

$(N+1) \times (N+1)$ linear system

$$\begin{bmatrix} K + \frac{N}{\gamma}\mathbb{I} & 1 \\ 1^\top & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ b \end{bmatrix} = \begin{bmatrix} \boldsymbol{y} \\ 0 \end{bmatrix}. \tag{4.4}$$

The resulting classifier has then the expression $f(\boldsymbol{x}) = \operatorname{sign}(\sum_{i=1}^{N} \alpha_i^\star k(\boldsymbol{x}, \boldsymbol{x_i}) + b^\star)$. The optimal hyperparameters $\sigma > 0$ and $\gamma > 0$ are estimated using grid search with validation on a hold-out set. The final classification is done by minimizing the Hamming distance on the one-versus-one outputs [SV99b]. In order to account for the amount of ink in the grey images $\boldsymbol{u}$ and $\boldsymbol{v}$, we also introduce a reweighted kernel that is defined as

$$k_{RW}(\boldsymbol{u}, \boldsymbol{v}) = \|\boldsymbol{u}\|_1 \|\boldsymbol{v}\|_1 k_W\left(\frac{\boldsymbol{u}}{\|\boldsymbol{u}\|_1}, \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|_1}\right). \tag{4.5}$$

Notice that a similar kernel has been defined with the Euclidean distance in [Mai16; CJM19].

In our experiments, we compare several methods based on k_W and k_{RW}, Wasserstein and Euclidean distances.

Core Wasserstein kernel

The "Core" method consists of solving (4.3) thanks to the feature map (4.2) associated to $K^{(\ell)}$. The parameter ℓ is selected such that all the selected eigenvalues are larger than 10^{-6} to avoid numerical instabilities. The optimal $\boldsymbol{w}^\star$ and $b^\star$ are then obtained by solving a linear system.

Core Wasserstein kernel with out-of-sample

Our second method named "Core + OOS" uses almost the same methodology as "Core". However, a subset of the training set is used to construct the truncated Wasserstein kernel of Proposition 4.2. Then the out-of-sample (OOS) formula (4.2) is used to construct an approximation of the kernel matrix on the full training dataset. The advantage of this approximation is that it can avoid the full eigendecomposition of the kernel matrix which is necessary for the "Core" method.

Indefinite Wasserstein kernel

For this third method, we simply use the indefinite Gram matrix associated to (4.5) for the kernel matrix and solve the system (4.4) associated to the dual formulation of LS-SVM.

While the associated optimization problem is not necessarily bounded in that case, the linear system (4.4) still has often a solution in practice. We name this method "Indefinite Wasserstein" in Fig. 4.5.

Gaussian RBF

The previous methods are compared with a classical LS-SVM classifier with kernel

$$k(\boldsymbol{u}, \boldsymbol{v}) = \exp\left(\frac{-\|\boldsymbol{u} - \boldsymbol{v}\|_2^2}{2\sigma^2}\right).$$

The parameters σ and γ are obtained by validation in the same way as above.

KNN

The same task is also performed for a kNN classifiers defined both with Euclidean and Wasserstein distances [SV]. Those two methods are considered as benchmarks to assess the accuracy of the kernel methods hereabove. Notice that the number of nearest neighbours k is selected by validation.

4.3.3 Description of the simulations

The simulations are repeated several times and the mean classification error rate is given as well as the standard deviation. We emphasize that the classes are balanced in each of the datasets. The code is provided on GitHub[2].

4.3.4 Discussion

The results obtained by classifiers defined with Wasserstein exponential kernel k_W outperform the Euclidean and Wasserstein kNN classifiers, as well as LS-SVM with a Gaussian RBF kernel (see Fig. 4.5 and Table 4.1). The latter is especially outperformed when the number of training data points is limited to a few thousands. We observed empirically that the advantage of k_W is indeed reduced as the size of the training set further increases. The reweighted version of the kernel k_{RW} also proves to be competitive probably because it incorporates the information about the amount of ink, which was suppressed in the normal k_W due to the normalization. However, the amount of ink seems to be a better class indicator in the Quickdraw dataset

[2] `https://github.com/hdeplaen/Wasserstein_Exponential_Kernels`

than in the two datasets consisting of digits. This counter-intuitive result may point towards the need for an alternative way of reincorporating the suppressed information due to the normalization. Surprisingly, the classifier obtained for the indefinite k_W kernel yields the best performance when the training set is larger. This observation certainly deserves further research. For moderate size training sets, LS-SVM classifiers can be competitive with respect to other methods that do not rely on convolutional neural networks. The latter are known to be performant for relatively large training datasets. While an advantage of Wasserstein based methods is an increased accuracy in the classification tasks of this paper, a main disadvantage is the increased training time.

4.4 Conclusion

In this paper, we proposed the use of Wasserstein squared exponential kernels for classifying shapes given relatively small training datasets. Although the computation of Wasserstein distances is expensive, it can be made possible thanks to the entropic regularization and the Sinkhorn algorithm, as it is well known. The so-called Wasserstein features are also proposed to serve as an approximation of the Wasserstein squared exponential kernel which is not necessarily positive semidefinite. In particular, this construction is possible if the bandwidth parameter is small enough as it is explained by elementary theoretical results. These theoretical results also open a door to more general exponential kernels based on any measure of similarity.

CHAPTER 5

Unbalanced Optimal Transport for Object Detection

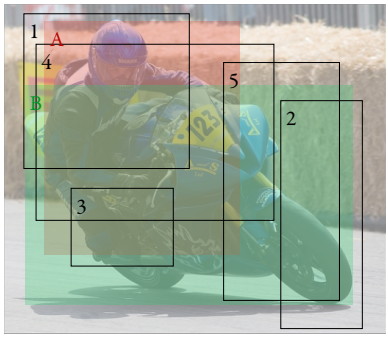
5.1 Introduction

Object detection models are in essence multi-task models, having to both localize objects in an image and classify them. In the context of supervised learning, each of these tasks heavily depends on a matching strategy. Indeed, determining which predicted object matches which ground truth object is a non-trivial yet essential task during the training (Figure 5.1a). In particular, the matching strategy must ensure that there is ideally exactly one prediction per ground truth object, at least during inference. Various strategies have emerged, often relying on hand-crafted components. They are proposed as scattered approaches that seem to have nothing in common, at least at first glance.

5.1.1 A Unifying Framework

To perform any match, a matching cost has to be determined. The example at Fig. 5.1b uses the *Generalized Intersection over Union* (GIoU) [RTG+19]. Given such a cost matrix, matching strategies include:

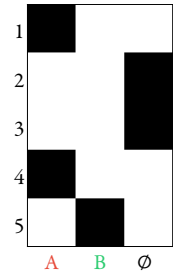
- Matching each prediction to the closest ground truth object. This often requires that the cost lies under a certain threshold [LAE+16; RHGS15; RDGF16; LGG+17], to avoid matching predictions that may be totally irrelevant for the current image. The disadvantage of this strategy is its redundancy: many predictions may point towards the



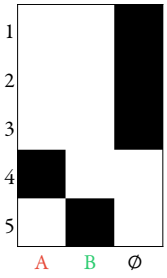
(a) Image №163 from the VOC training dataset. The ground truth boxes are colored, and the predictions are outlined in black.

	A	B	∅
1	0.61	0.96	0.8
2	1.34	0.88	0.8
3	0.88	0.89	0.8
4	0.4	0.65	0.8
5	1.11	0.73	0.8

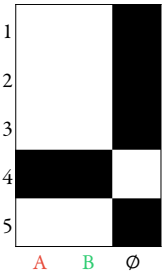
(b) Costs between the predictions and the ground truth ($1 - \text{GIoU}$). The background cost is $c_\varnothing = 0.8$.



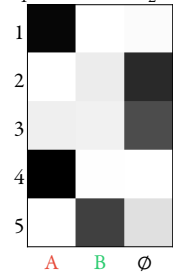
(c) Prediction to best ground truth (Unbalanced OT with $\epsilon = 0, \tau_1 \to +\infty$ and $\tau_2 = 0$).



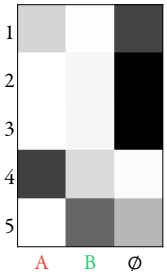
(d) Hungarian matching (OT with $\epsilon = 0, \tau_1 \to +\infty$ and $\tau_2 \to +\infty$).



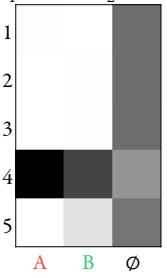
(e) Ground truth to best prediction (Unbalanced OT with $\epsilon = 0, \tau_1 = 0$ and $\tau_2 \to +\infty$).



(f) Unbalanced OT with $\epsilon = 0.05, \tau_1 = 100$ and $\tau_2 = 0.01$.



(g) OT with $\epsilon = 0.05$ ($\tau_1 \to +\infty$ and $\tau_2 \to +\infty$).



(h) Unbalanced OT with $\epsilon = 0.05, \tau_1 = 0.01$ and $\tau_2 = 100$.

FIGURE 5.1: Different matching strategies. All are particular cases of *Unbalanced Optimal Transport*. A match ($\hat{P}_{i,j} = 1$) is denoted by a black square and it is white if there is no match ($\hat{P}_{i,j} = 0$).

same ground truth object. In Fig. 5.1c, both predictions 1 and 4 are matched towards ground truth object A. Furthermore, some ground truth objects may be unmatched. A solution to this is to increase the number of predicted boxes drastically. This is typically the case with anchors boxes and region proposal methods.

- The opposite strategy is to match each ground truth object to the best prediction [HZRS15; LAE+16]. This ensures that there is no redundancy and every ground truth object is matched. This also comes with the opposite problem: multiple ground truth objects may be matched to the same prediction. In Fig. 5.1e, both ground truth objects A and B are matched to prediction 4. This can be mitigated by having more predictions, but then many of those are left unmatched, slowing convergence [LAE+16].

- A compromise is to perform a *Bipartite Matching* (BM), using the *Hungarian algorithm* [Kuh55; Mun57], for example [CMS+20; ZSL+21]. The matching is one-to-one, minimizing the total cost (Definition 5.2). Every ground truth object is matched to a unique prediction, thus reducing the number of predictions needed, as shown in Fig. 5.1d. A downside is that the one-to-one matches may vary from one epoch to the next, again slowing down convergence [LZL+22]. This strategy is difficult to parallelize, *i. e.* to take advantage of GPU architectures.

All of these strategies have different properties and it seems that one must choose either one or the other, optionally combining them using savant heuristics [LAE+16]. There is a need for a unifying framework. As we show in this paper, *Unbalanced Optimal Transport* [CPSV18b] offers a good candidate for this (Figure 5.1). It not only unifies the different strategies here above, but also allows to explore all cases in between. The cases presented in Figures 5.1c, 5.1d and 5.1e correspond to the limit cases. This opens the door for all intermediate settings. Furthermore, we show how regularizing the problem induces smoother matches, leading to faster convergence of DETR, avoiding the problem described for the BM. In addition, the particular choice of entropic regularization leads to a class of fast parallelizable algorithms on GPU known as *scaling algorithms* [Cut13b; CPSV18a], of which we provide a compiled implementation on GPU. Our code and additional resources are publicly available[1].

[1] `https://hdeplaen.github.io/uotod`

5.1.2 Related Work

Matching Strategies

Most *two-stage* models often rely on a huge number of initial predictions, which is then progressively reduced in the region proposal stage and refined in the classification stage. Many different strategies have been proposed for the initial propositions and subsequent reductions, ranging from training no deep learning networks [GDDM14], to only train those for the propositions [Gir15; LDG+17; HZRS15], to training networks for both propositions and reductions [RHGS15; PCS+19; HGDG17; CV18; DLHS16]. Whenever a deep learning network is trained, each prediction is matched to the closest ground truth object provided it lies beneath a certain threshold. Moreover, the final performance of these models heavily depends on the hand-crafted anchors [LOW+20].

Many *one-stage* models rely again on predicting a large number of initial predictions or *anchor boxes*, covering the entire image. As before, each anchor box is matched towards the closest ground truth object with certain threshold constraints [RDGF16; LGG+17]. In [LAE+16], this is combined with matching each ground truth object to the closest anchor box and a specific ratio heuristic between the matched and unmatched predictions. The matching of the fixed anchors is justified to avoid a collapse of the predictions towards the same ground truth objects. Additionally, this only works if the number of initial predictions is sufficiently large to ensure that every ground truth object is matched by at least one prediction. Therefore, it requires further heuristics, such as *Non-Maximal Suppression* (NMS) to guarantee a unique prediction per ground truth object, at least during the inference.

By using the *Hungarian algorithm*, DETR [CMS+20] removed the need for a high number of initial predictions. The matched predictions are improved with a multi-task loss, and the remaining predictions are trained to predict the background class $\emptyset$. Yet, the model converges slowly due to the instability of BM, causing inconsistent optimization goals at early training stages [LZL+22]. Moreover, the sequential nature of the Hungarian algorithm does not take full advantage of the GPU architecture. Several subsequent works accelerate the convergence of DETR by improving the architecture of the model [ZSL+21; LLZ+21] and by adding auxiliary losses [LZL+22], but not by exploring the matching procedure.

Optimal Transport

OT emerges from an old problem [Mon81], relaxed by a newer formulation [Kan58]. It gained interest in the machine learning community since the re-discovery of *Sinkhorn's algorithm* [Cut13b] and opened the door for improvements in a wide variety of applica-

tions ranging from graphical models [MMC16], kernel methods [KZR16; DFS20], loss design [FZM+15], auto-encoders [TBGS18; KPMR18; RST18] or generative adversarial networks [ACB17; GAA+17].

More recent incursions in computer vision have been attempted, *e. g.* for the matching of predicted classes [HLS+20], a loss for rotated object boxes [YYM+21] or a new metric for performance evaluation [OTN+22]. Considering the matching of predictions to ground truth objects, recent attempts using OT bare promising results [GLL+21; GLW+21]. However, when the *Hungarian algorithm* is mentioned, it is systematically presented in opposition to OT [GLL+21; VJ22]. We lay a rigorous connection between those two approaches in computer vision.

Unbalanced OT has seen a much more recent theoretical development [CPSV18b; Chi17]. The hard mass conservation constraints in the objective function are replaced by soft penalization terms. Its applications are scarcer, but we must mention here relatively recent machine learning applications in motion tracking [LBR20] and domain adaptation [FSFC21].

5.1.3 Contributions

1. We propose a unifying matching framework based on *Unbalanced Optimal Transport*. It encompasses both the *Hungarian algorithm*, the matching of the predictions to the closest ground truth boxes and the ground truth boxes to the closest predictions;
2. We show that these three strategies correspond to particular limit cases and we subsequently present a much broader class of strategies with varying properties;
3. We demonstrate how entropic regularization can speed up the convergence during training and additionally take advantage of GPU architectures;
4. We justify the relevancy of our framework by exploring its interaction with NMS and illustrate how it is on par with the state-of-the-art.

5.1.4 Notations and Definitions

Notations

Throughout the paper, we use small bold letters to denote a vector $\boldsymbol{a} \in \mathbb{R}^N$, with elements $a_i \in \mathbb{R}$. Similarly, matrices are denoted by bold capital letters such as $\boldsymbol{A} \in \mathbb{R}^{N \times M}$, with elements $A_{i,j} \in \mathbb{R}$. The notation 1_N represents a column-vector of ones, of size N, and

$1_{N\times M}$ the matrix equivalent of size $N \times M$. The identity matrix of size N is $\boldsymbol{I}_{N,N}$. With $N = \{1, 2, \dots, N\}$, we denote the set of integers from 1 to N. The probability simplex uses the notation $\Delta^N = \left\{\boldsymbol{u} \in \mathbb{R}^N_{\geq 0} \middle| \sum_i u_i = 1\right\}$ and represents the set of discrete probability distributions of dimension N. This extends to the set of discrete joint probability distributions $\Delta^{N\times M}$.

Definitions

For each image, the set $\{\hat{\boldsymbol{y}}_i\}_{i=1}^{N_p}$ denotes the predictions and $\{\boldsymbol{y}_j\}_{j=1}^{N_g}$ the ground truth samples. Each ground truth sample combines a target class and a bounding box position: $\boldsymbol{y}_j = \left[\boldsymbol{c}_j, \boldsymbol{b}_j\right] \in \mathbb{R}^{N_c+4}$ where $\boldsymbol{c}_j \in \{0, 1\}^{N_c}$ is the target class in one-hot encoding with N_c the number of classes and $\boldsymbol{b}_j \in [0, 1]^4$ defines the relative bounding box center coordinates and dimensions. The predictions are defined similarly $\hat{\boldsymbol{y}}_i = [\,\hat{\boldsymbol{c}}_i, \hat{\boldsymbol{b}}_i\,] \in \mathbb{R}^{N_c+4}$, but the predicted classes may be non-binary $\hat{\boldsymbol{c}}_i \in [0, 1]^{N_c}$. Sometimes, predictions are defined relatively to fixed anchor boxes $\tilde{\boldsymbol{b}}_i$.

5.2 Optimal Transport

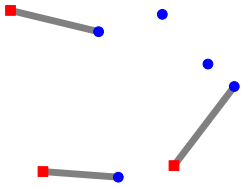
In this section, we show how *Optimal Transport* and then its *Unbalanced* extension unify both the *Hungarian algorithm* used in DETR [CMS+20], and matching each prediction to the closest ground truth object used in both Faster R-CNN [RHGS15] and SSD [LAE+16]. We furthermore stress the advantages of entropic regularization, both computationally and qualitatively. This allows us to explore a new continuum of matching methods, with varying properties.

Definition 5.1 (Optimal Transport)**.** *Given a distribution $\boldsymbol{\alpha} \in \Delta^{N_p}$ associated to the predictions $\{\hat{\boldsymbol{y}}_i\}_{i=1}^{N_p}$, and another distribution $\boldsymbol{\beta} \in \Delta^{N_g}$ associated with the ground truth objects $\{\boldsymbol{y}_j\}_{j=1}^{N_g}$. Let us consider a pair-wise matching cost $\mathcal{L}_{match}(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j)$ between a prediction $\hat{\boldsymbol{y}}_i$ and a ground truth object $\boldsymbol{y}_j$. We now define* Optimal Transport (OT) *as finding the match $\boldsymbol{P}$ that minimizes the following problem:*

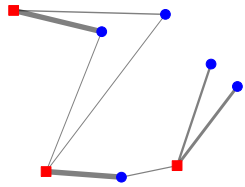
$$\hat{\boldsymbol{P}} = \underset{\boldsymbol{P} \in \mathcal{U}(\boldsymbol{\alpha}, \boldsymbol{\beta})}{\arg\min} \left\{ \sum_{i,j=1}^{N_p, N_g} P_{i,j} \mathcal{L}_{match}\left(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j\right) \right\}, \tag{5.1}$$

with transport polytope (the set of all admissible solutions)

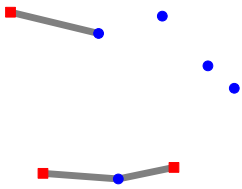
$$\mathcal{U}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \left\{ \boldsymbol{P} \in \mathbb{R}^{N_p \times N_g}_{\geq 0} : \sum_{j=1}^{N_g} P_{i,j} = \alpha_i, \sum_{i=1}^{N_p} P_{i,j} = \beta_j \right\}. \tag{5.2}$$



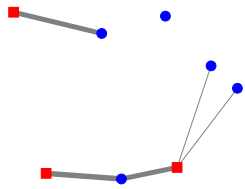
(a) BM as a particular case of OT with no regularization ($\varepsilon = 0$). The *Hungarian algorithm* obtains the same solution.



(b) OT with regularization ($\varepsilon \neq 0$). The regularization smoothens the matching allowing for multiple connections.The problem here is fully balanced ($\tau_1, \tau_2 \to \infty$).



(c) Unbalanced OT with regularization ($\varepsilon \neq 0$ and $\tau_1 \ll \tau_2$). The smoothing is also visible.



(d) Matching each ground truth object to the closest prediction as Unbalanced OT without regularization with $\varepsilon = 0, \tau_1 = 0$ and $\tau_2 \to \infty$.

Figure 5.2: Example of the influence of the parameters. The blue dots represent predictions $\hat{\boldsymbol{y}}_i$. The red squares represent ground truth objects $\boldsymbol{y}_j$. The distributions $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are defined as in Prop. 5.1. The thickness of the lines is proportional to the amount transported $P_{i,j}$. Only sufficiently thick lines are plotted. The dummy *background* ground truth $\boldsymbol{y}_{N_g+1} = \emptyset$ is not shown, nor are the connections to it. We can see that the plots a and b are balanced and the mass constraints are respected ($\boldsymbol{P} \in \mathcal{U}(\boldsymbol{\alpha}, \boldsymbol{\beta})$). This can be seen graphically by noticing that the thickness of the lines sum up to the same mass for each red point.

Provided that certain conditions apply to the underlying cost $\mathscr{L}_{\text{match}}$, the minimum defines a distance between $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, referred to as the Wasserstein distance $\mathscr{W}(\boldsymbol{\alpha}, \boldsymbol{\beta})$ (for more information, we refer to monographs [Vil09; San15; PC+19]; see also Appendix **??**).

5.2.1 The Hungarian Algorithm

The Hungarian algorithm solves the *Bipartite Matching* (BM). We will now show how this is a particular case of Optimal Transport.

Definition 5.2 (Bipartite Matching)**.** *Given the same objects as in Definition 5.1, the* Bipartite Matching (BM) *minimizes the cost of the pairwise matches between the ground truth objects with the predictions:*

$$\hat{\sigma} = \arg\min \left\{ \sum_{j=1}^{N_g} \mathscr{L}_{match} \left(\hat{\boldsymbol{y}}_{\sigma(j)}, \boldsymbol{y}_j \right) : \sigma \in \mathscr{P}_{N_g}(N_p) \right\}, \tag{5.3}$$

where $\mathscr{P}_{N_g}(N_p) = \left\{ \sigma \in \mathscr{P}(N_p) \,\middle|\, |\sigma| = N_g \right\}$ *is the set of possible combinations of* N_g *in* N_p*, with* $\mathscr{P}(N_p)$ *the power set of* N_p *(the set of all subsets).*

BM tries to assign each ground truth $\boldsymbol{y}_j$ to a different prediction $\hat{\boldsymbol{y}}_i$ in a way to minimize the total cost. In contrast to OT, BM does not consider any underlying distributions $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, all ground truth objects and predictions are implicitly considered to be of same mass. Furthermore, it only allows one ground truth to be matched to a unique prediction, some of these predictions being left aside and matched to nothing (which is then treated as a matching to the background $\emptyset$). The OT must match all ground truth objects to all predictions, not allowing any predictions to be left aside. However, the masses of the ground truth objects are allowed to be split between different predictions and inversely, as long as their masses correctly sum up ($\boldsymbol{P} \in \mathcal{U}(\boldsymbol{\alpha}, \boldsymbol{\beta})$).

Particular Case of OT

A solution for an imbalanced number of predictions compared to the number of ground truth objects would be to add dummy ground truth objects—the background $\emptyset$—to even the balance. Concretely, one could add a new ground truth $\boldsymbol{y}_{N_g+1} = \emptyset$, with the mass equal to the unmatched number of predictions. In fact, doing so directly results in performing a BM.

Proposition 5.1. *The Hungarian algorithm with* N_p *predictions and* $N_g \leq N_p$ *ground truth objects is a particular case of OT with* $\boldsymbol{P} \in \mathcal{U}(\boldsymbol{\alpha}, \boldsymbol{\beta}) \subset \mathbb{R}^{N_p \times (N_g+1)}$*, consisting of the predictions and the ground truth objects, with the background added* $\{\boldsymbol{y}_j\}_{j=1}^{N_g+1} = \{\boldsymbol{y}_j\}_{j=1}^{N_g} \cup \left(\boldsymbol{y}_{N_g+1} = \emptyset\right)$*. The chosen underlying distributions are*

$$\boldsymbol{\alpha} = \frac{1}{N_p} [\underbrace{1,\ 1,\ 1,\ \ldots,\ 1}_{N_p \text{ predictions}}], \tag{5.4}$$

$$\boldsymbol{\beta} = \frac{1}{N_p} [\underbrace{1,\ 1,\ \ldots,\ 1}_{N_g \text{ ground truth objects}},\ \underbrace{(N_p - N_g)}_{\text{background } \emptyset}], \tag{5.5}$$

provided the background cost is constant: $\mathscr{L}_{match}\left(\hat{\boldsymbol{y}}_i, \emptyset\right) = c_\emptyset$. *In particular for* $j \in N_g$, *we have* $\hat{\sigma}(j) = \left\{i : P_{i,j} \neq 0\right\}$, *or equivalently* $\hat{\sigma}(j) = \left\{i : P_{i,j} = 1/N_p\right\}$.

Proof. We refer to Appendix **??**. ■

In other words, we can read the matching to each ground truth in the columns of $\hat{\boldsymbol{P}}$. The last columns represents all the predictions matched to the background $\hat{\sigma}(N_g + 1)$. Alternatively and equivalently, we can read the matching of each prediction i in the rows, the ones being matched to the background have a $\hat{P}_{i,N_g+1} = 1/N_p$.

Solving the Problem

Both OT and BM are linear programs. Using generic formulations would lead to a $\left(N_p + N_g + 1\right) \times N_p\left(N_g + 1\right)$ equality constraint matrix. It is thus better to exploit the particular bipartite structure of the problem. In particular, two families of algorithms have emerged: *Dual Ascent Methods* and *Auction Algorithms* [PC+19]. The Hungarian algorithm is a particular case of the former and classically runs with an $\mathcal{O}\left(N_p^4\right)$ complexity [Mun57], further reduced to cubic by [EK72]. Although multiple GPU implementations of a BM solver have been proposed [VR09; DN16; FB12], the problem remains poorly parallelizable because of its sequential nature. To allow for efficient parallelization, we must consider a slightly amended problem.

5.2.2 Regularization

We show here how we can replace the *Hungarian algorithm* by a class of algorithms well-suited for parallelization, obtained by adding an entropy regularization.

Definition 5.3 (OT with regularization)**.** *We consider a regularization parameter* $\epsilon \in \mathbb{R}_{\geq 0}$. *Extending Definition 5.1 (OT), we define the* Optimal Transport with regularization *as the following minimization problem:*

$$\hat{\boldsymbol{P}} = \underset{\boldsymbol{P} \in \mathcal{U}(\boldsymbol{\alpha}, \boldsymbol{\beta})}{\arg\min} \left\{ \sum_{i,j=1}^{N_p, N_g} P_{i,j} \mathscr{L}_{match}\left(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j\right) - \epsilon\, \mathrm{H}(\boldsymbol{P}) \right\}, \tag{5.6}$$

with $\mathrm{H} : \Delta^{N \times M} \to \mathbb{R}_{\geq 0} : \boldsymbol{P} \mapsto -\sum_{i,j} P_{i,j}(\log(P_{i,j}) - 1)$ *the entropy of the match* $\boldsymbol{P}$, *with* $0 \ln(0) = 0$ *by definition.*

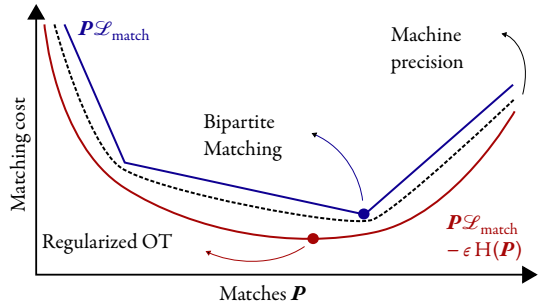


FIGURE 5.3: Effect of the regularization on the minimization of the matching cost. The red line corresponds to the regularized problem ($\epsilon \neq 0$) and the blue to the unregularized one ($\epsilon = 0$).

Sinkhorn's Algorithm

The entropic regularization used when finding the match $\hat{\boldsymbol{P}}$ ensures that the problem is smooth for $\epsilon \neq 0$ (see Figure 5.3). The advantage is that it can now be solved very efficiently using *scaling algorithms* and in this particular case the algorithm of *Sinkhorn*. It is particularly suited for parallelization [Cut13b], with some later speed refinements [ANR17b; ABGR19]. Reducing the regularization progressively renders the scaling algorithms numerically unstable, although some approaches have been proposed to reduce the regularization further by working in log-space [Sch19; CPSV18a]. In the limit of $\epsilon \to 0$, we recover the exact OT (Definition 5.1) and the scaling algorithms cannot be used anymore. Parallelization is lost and we must resolve to use the sequential algorithms developed in Section 5.2.1. In brief, regularization allows to exploit GPU architectures efficiently, whereas the Hungarian algorithm and similar cannot.

Smoother Matches

When no regularization is used as in the Hungarian algorithm, close predictions and ground truth objects can exchange their matches from one epoch to the other, during the training. This causes a slow convergence of DETR in the early stages of the training [LZL+22]. The advantage of the regularization not only lies in the existence of efficient algorithms but also allows for a reduction of sparsity. This results in a less drastic match than the Hungarian algorithm obtains. A single ground truth could be matched to multiple predictions and inversely. The proportion of these multiple matches is controlled by the regularization parameter ϵ. An illustration can be found in Figures 5.2a and 5.2b.

5.2.3 Unbalanced Optimal Transport

We will now show how considering soft constraints instead of hard leads to an even greater generalization of the various matching techniques used in object detection models. In particular, matching each prediction to the closest ground truth is a limit case of the *Unbalanced OT*.

Definition 5.4 (Unbalanced OT)**.** *We consider two constraint parameters* $\tau_1, \tau_2 \in \mathbb{R}_{\geq 0}$. *Extending Definition 5.3 (OT with regularization), we define the* Unbalanced OT with regularization *[CPSV18a] as the following minimization problem:*

$$\hat{\boldsymbol{P}} = \underset{\boldsymbol{P} \in \mathbb{R}_{\geq 0}^{N_p \times N_g}}{\arg\min} \Big\{ \epsilon \, \mathrm{KL}(\boldsymbol{P} \,\|\, \boldsymbol{K}_\epsilon) + \tau_1 \mathrm{KL}(\boldsymbol{P} 1_{N_g} \,\|\, \boldsymbol{\alpha}) + \tau_2 \mathrm{KL}(1_{N_p}^\top \boldsymbol{P} \,\|\, \boldsymbol{\beta}) \Big\}, \tag{5.7}$$

where $\mathrm{KL} : \mathbb{R}_{\geq 0}^{N \times M} \times \mathbb{R}_{> 0}^{N \times M} \to \mathbb{R}_{\geq 0} : (\boldsymbol{U}, \boldsymbol{V}) \mapsto \sum_{i,j=1}^{N \times M} U_{i,j} \log(U_{i,j}/V_{i,j}) - U_{i,j} + V_{i,j}$ *is the* Kullback-Leibler divergence – *also called* relative entropy – *between matrices or vectors when* $M = 1$, *with* $0 \ln(0) = 0$ *by definition. The Gibbs kernel* $\boldsymbol{K}_\epsilon$ *is given by* $(K_\epsilon)_{i,j} = \exp\left(-\mathscr{L}_{match}\left(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j\right)/\epsilon\right)$.

We can see by development that the first term corresponds to the matching term $\boldsymbol{P}\mathscr{L}_{\text{match}}$ and an extension of the entropic regularization term $\mathrm{H}(\boldsymbol{P})$. The two additional terms replace the transport polytope's hard constraints $\mathscr{U}(\boldsymbol{\alpha}, \boldsymbol{\beta})$ that required an exact equality of mass for both marginals. These new soft constraints allow for a more subtle sensitivity to the mass constraints as it allows to slightly diverge from them. It is clear that in the limit of $\tau_1, \tau_2 \to +\infty$, we recover the "balanced" problem (Definition 5.3). This definition naturally also defines Unbalanced OT without regularization if $\epsilon = 0$. The matching term would remain and the entropic one disappear.

Matching to the Closest

Another limit case is however particularly interesting in the quest for a unifying framework of the matching strategies. If the mass constraint is to be perfectly respected for the predictions ($\tau_1 \to \infty$), but not at all for the ground truth objects ($\tau_2 = 0$), it suffices to assign the closest ground truth to each prediction. The same ground truth object could be assigned to multiple predictions and another could not be matched at all, not respecting the hard constraint for the ground truth $\boldsymbol{\beta}$. Each prediction however is exactly assigned once, perfectly respecting

the mass constraint for the predictions $\boldsymbol{\alpha}$. By assigning a low enough value to the background, a prediction would be assigned to it provided all the other ground truth objects are further. In other words, the background cost would play the role of a *threshold* value.

Proposition 5.2 (Matching to the closest)**.** *We consider the same objects as Proposition 5.1. In the limit of* $\tau_1 \to \infty$ *and* $\tau_2 = 0$*, Unbalanced OT (Definition 5.4) without regularization* ($\varepsilon = 0$) *admits as solution each prediction being matched to the closest ground truth object unless that distance is greater than a threshold value* $\mathcal{L}_{match}\left(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_{N_g+1} = \emptyset\right) = c_\emptyset$*. It is then matched to the background* $\emptyset$*. In particular, we have*

$$\hat{P}_{i,j} = \begin{cases} \frac{1}{N_p} & \text{if } j = \arg\min_{j \in N_g+1} \left\{\mathcal{L}_{match}\left(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j\right)\right\}, \\ 0 & \text{otherwise.} \end{cases} \tag{5.8}$$

Proof. We refer to Appendix **??**. ■

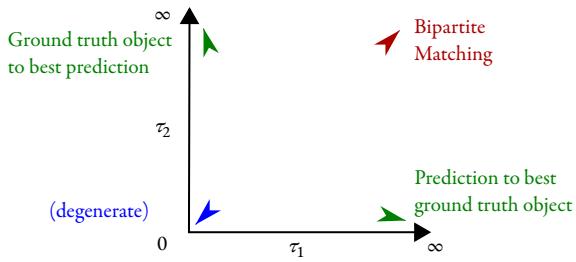


FIGURE 5.4: Limit cases of Unbalanced OT without regularization ($\varepsilon = 0$).

The converse also holds. If the ground truth objects mass constraints were to be perfectly respected ($\tau_2 \to \infty$), but not the predictions ($\tau_1 \to 0$), each ground truth would then be matched to the closest prediction. The background would be matched to the remaining predictions. Some predictions could not be matched and other ones multiple times. The limits of Unbalanced OT are illustrated in Fig. 5.4. By setting the threshold sufficiently high, we get an exact minimum, i.e., where every prediction is matched to the closest ground truth. This can be observed in Figure 5.2d.

Scaling Algorithm

Similarly as before, adding entropic regularization ($\varepsilon \neq 0$) to the *Unbalanced OT* allows it to be solved efficiently on GPU with a scaling algorithm, as an extension of Sinkhorn's algorithm [CPSV18a; Chi17]. The regularization still also allows for smoother matches, as shown in Figure 5.2c.

Softmax

In the limit of $\tau_1 \to +\infty$ and $\tau_2 = 0$, the solution corresponds to a softmax over the ground truth objects for each prediction. The regularization ε controls then the "softness" of the softmax, with $\varepsilon = 1$ corresponding to the conventional softmax and $\varepsilon \to 0$ the matching to the closest. We refer to Appendix **??** for more information.

5.3 Matching

Following previous work [CMS+20; ZSL+21; RHGS15; RDGF16; LAE+16], we define a multi-task matching cost between a prediction $\hat{\boldsymbol{y}}_i$ and a ground truth object $\boldsymbol{y}_j$ as the composition of a classification loss ensuring that similar object classes are matched together and a localization loss ensuring the correspondence of the positions and shapes of the matched boxes $\mathscr{L}(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j) = \mathscr{L}_{\text{classification}}(\hat{\boldsymbol{c}}_i, \boldsymbol{c}_j) + \mathscr{L}_{\text{localization}}(\hat{\boldsymbol{b}}_i, \boldsymbol{b}_j)$. Most models, however, do not use the same loss to determine the matches as the one used to train the model. We therefore refer to these two losses as $\mathscr{L}_{\text{match}}$ and $\mathscr{L}_{\text{train}}$. The training procedure is the following: first find a match $\hat{\boldsymbol{P}}$ given a matching strategy and matching cost $\mathscr{L}_{\text{match}}$, then compute the loss $N_p \sum_{i=1}^{N_p} \sum_{j=1}^{N_g} \hat{P}_{ij} \mathscr{L}_{\text{train}}(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j)$ where the particular training loss for the background ground truth includes only a classification term $\mathscr{L}_{\text{train}}(\hat{\boldsymbol{y}}_i, \emptyset) = \mathscr{L}_{\text{classification}}(\hat{\boldsymbol{c}}_i, \emptyset)$.

5.3.1 Detection Transformer (DETR)

The object detection is performed by matching the predictions to the ground truth boxes with the *Hungarian algorithm* applied to the loss $\mathscr{L}_{\text{match}}(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j) = \lambda_{\text{prob}}(1 - \langle \hat{\boldsymbol{c}}_i, \boldsymbol{c}_j \rangle) + \lambda_{\ell^1} \|\hat{\boldsymbol{b}}_i - \boldsymbol{b}_j\|_1 + \lambda_{\text{GIoU}}(1 - \text{GIoU}(\hat{\boldsymbol{b}}_i, \boldsymbol{b}_j))$ (Definition 5.2). To do so, the number of predictions and ground truth boxes must be of the same size. This is achieved by padding the ground truths with $(N_p - N_g)$ dummy *background* $\emptyset$ objects. Essentially, this is the same as what is developed in Proposition 5.1. The obtained match is then used to define an object-specific loss, where each matched prediction is pushed toward its corresponding ground truth object. The predictions that are not matched to a ground truth object are considered to be matched with the background and are pushed to predict the background class. The training loss uses the cross-entropy (CE) for classification: $\mathscr{L}_{\text{train}}(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j) = \lambda_{\text{CE}} \mathscr{L}_{\text{CE}}(\hat{\boldsymbol{c}}_i, \boldsymbol{c}_j) + \lambda_{\ell^1} \|\hat{\boldsymbol{b}}_i - \boldsymbol{b}_j\|_1 + \lambda_{\text{GIoU}}(1 - \text{GIoU}(\hat{\boldsymbol{b}}_i, \boldsymbol{b}_j))$. By directly applying Proposition 5.1 and adding entropic regularization (Definition 5.3), we can use *Sinkhorn's algorithm* and push each prediction $\hat{\boldsymbol{y}}_i$ to ground truth $\boldsymbol{y}_j$ according to weight $\hat{P}_{i,j}$. In particular, for any non-zero $\hat{P}_{i,N_g+1} \neq 0$, the prediction $\hat{\boldsymbol{y}}_i$ is pushed toward the background $\boldsymbol{y}_{N_g+1} = \emptyset$ with weight $\hat{P}_{i,N_g+1}$.

5.3.2 Single Shot MultiBox Detector (SSD)

The Single Shot MultiBox Detector [LAE+16] uses a matching cost only comprised of the IoU between the fixed anchor boxes $\tilde{\boldsymbol{b}}_i$ and the ground truth boxes: $\mathscr{L}_{\text{match}}(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j) = 1 - \text{IoU}(\tilde{\boldsymbol{b}}_i, \boldsymbol{b}_j)$ (the GIoU was not published yet [RTG+19]). Each ground truth is first matched toward the closest anchor box. Anchor boxes are then matched to a ground truth object if the matching cost is below a threshold of 0.5. In our framework, this corresponds to applying $\tau_1 = 0$ and $\tau_2 \to \infty$ for the first phase and then $\tau_1 \to \infty$ and $\tau_2 = 0$ with $c_\emptyset = 0.5$ (see Proposition 5.2). Here again, by adding entropic regularization (Definition 5.4), we can solve this using a *scaling algorithm*. We furthermore can play with the parameters τ_1 and τ_2 to make the matching tend slightly more towards a matching done with the *Hungarian algorithm* (Figure 5.2). Again, the training uses a different loss than the matching, in particular $\mathscr{L}_{\text{train}}(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_j) = \lambda_{\text{CE}}\mathscr{L}_{\text{CE}}(\hat{\boldsymbol{c}}_i, \boldsymbol{c}_j) + \lambda_{\text{smooth }\ell^1}\mathscr{L}_{\text{smooth }\ell^1}(\hat{\boldsymbol{b}}_i, \boldsymbol{b}_j)$.

Hard Negative Mining

Instead of using all negative examples $N_{\text{neg}} = (N_p - N_g)$ (predictions matched to background), the method sorts them using the highest confidence loss $\mathscr{L}_{\text{CE}}(\hat{\boldsymbol{c}}_i, \emptyset)$ and picks the top ones so that the ratio between the hard negatives and positives $N_{\text{pos}} = N_g$ is at most 3 to 1. Since $\hat{\boldsymbol{P}}$ is non-binary, we define the number of negatives and positives to be the sum of the matches to the background $N_{\text{neg}} = N_p \sum_{i=1}^{N_p} \hat{P}_{i,(N_g+1)}$ and to the ground truth objects $N_{\text{pos}} = N_p \sum_{j=1}^{N_g} \sum_{i=1}^{N_p} \hat{P}_{ij}$. We verify that for any $\boldsymbol{P} \in \mathscr{U}(\boldsymbol{\alpha}, \boldsymbol{\beta})$, we have the same number of positives and negatives as the initial model: $N_{\text{neg}} = (N_p - N_g)$ and $N_{\text{pos}} = N_g$. Hence, hard negatives are the K predictions with the highest confidence loss $\hat{P}_{k,(N_g+1)}\mathscr{L}_{\text{CE}}(\hat{\boldsymbol{c}}_k, \emptyset)$ such that the mass of kept negatives is at most triple the number of positives: $N_p \sum_{k=1}^{K} \hat{P}^s_{k,(N_g+1)} \leq 3N_{\text{pos}}$, where $\hat{\boldsymbol{P}}^s$ is a permutation of transport matrix $\hat{\boldsymbol{P}}$ with rows sorted by highest confidence loss.

5.4 Experimental Results & Discussion

We show that matching based on *Unbalanced Optimal Transport* generalizes many different matching strategies and performs on par with methods that use either *Bipartite Matching* or anchor boxes along with matching each prediction to the closest ground truth box with a threshold. We then analyze the influence of constraint parameter τ_2 by training SSD with and without NMS for multiple parameter values. Finally, we show that OT with entropic

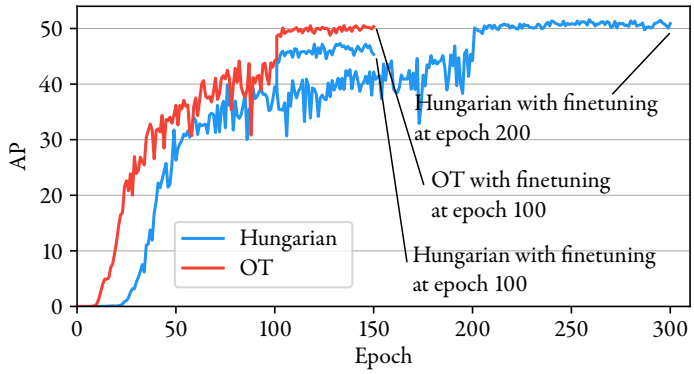


Figure 5.5: Convergence curves for DETR on the Color Boxes dataset. The model converges faster with a regularized matching.

regularization both improves the convergence and is faster to compute than the Hungarian algorithm in case of many matches.

5.4.1 Setup

Datasets

We perform experiments on a synthetic object detection dataset with 4.800 training and 960 validation images and on the large-scale COCO [LMB+14] dataset with 118,287 training and 5,000 validation test images. We report on mean Average Precision (AP) and mean Average Recall (AR). The two metrics are an average of the per-class metrics following COCO's official evaluation procedure. For the Color Boxes synthetic dataset, we uniformly randomly draw between 0 and 30 rectangles of 20 different colors from each image. Appendix A.5 provides the detailed generation procedure and sample images.

Training

For a fair comparison, the classification and localization costs for matching and training are identical to the ones used by the models. Unless stated otherwise, we train the models with their default hyper-parameter sets. DETR and Deformable DETR are trained with hyper-parameters $\lambda_{\text{prob}} = \lambda_{\text{CE}} = 2$, $\lambda_{\ell^1} = 5$ and $\lambda_{\text{GIoU}} = 2$. For Deformable DETR, we found the classification cost to be overwhelmed by the localization costs in the regularized

	Model	Matching	τ_2	Epochs	AP	AR
Color Boxes	DETR	Hungarian	(∞)	300	**50.9**	**65.7**
	DETR	Hungarian	(∞)	150	45.3	60.7
	DETR	OT	(∞)	**150**	50.3	**65.7**
	D. DETR	Hungarian	(∞)	50	**64.0**	75.9
	D. DETR	OT	(∞)	50	63.5	**76.5**
COCO	D. DETR	Hungarian	(∞)	50	**44.5**	**63.0**
	D. DETR	OT	(∞)	50	44.2	62.0
	SSD300	Two Stage	—-	120	**24.9**	**36.8**
	SSD300	Unb. OT	0.01	120	24.7	36.4

Table 5.1: Object detection metrics for different models and loss functions on the Color Boxes and COCO datasets.

minimization problem (Definition 5.3). We therefore set $\lambda_{\text{prob}} = 5$. We, however keep $\lambda_{\text{CE}} = 2$ so that the final loss value for a given matching remains unchanged. SSD is trained with original hyper-parameters $\lambda_{\text{CE}} = \lambda_{\text{smooth } \ell^1} = 1$. For OT, we set the entropic regularization to $\epsilon = \epsilon_0 / (\log(2N_p) + 1)$ where $\epsilon_0 = 0.12$ for all models (App. **??**). In the following experiments, the Unbalanced OT is solved with multiple values of τ_2 whereas τ_1 is fixed to a large value $\tau_1 = 100$ to simulate a hard constraint. In practice, we limit the number of iterations of the scaling algorithm. This provides a good enough approximation [GLW+21].

5.4.2 Timing Analysis for SSD

As can be seen in Table 5.2, OT-based matches improve the epoch time (forward pass, compute the match cost, matching algorithm, and backward pass; in blue) for SSD with the Hungarian algorithm by almost 50%. The difference is smaller for DETR and variants as the models are proportionally heavier and the number of predictions smaller.

5.4.3 Unified Matching Strategy

DETR and Deformable DETR

Convergence curves for DETR on the Color Boxes dataset are shown in Fig. 5.5 and associated metrics are presented in Table 5.1. DETR converges in half the number of epochs with the regularized balanced OT formulation. This confirms that one reason for slow DETR convergence is the discrete nature of BM, which is unstable, especially in the early stages

Epoch step	OT	Unb. OT	Hung.	2-step
Preprocessing	6.3 ms	*idem*	*idem*	*idem*
Forward pass	5.8 ms	*idem*	*idem*	*idem*
Anchor gen.	54.2 ms	*idem*	*idem*	*idem*
Match cost	4.2 ms	*idem*	*idem*	*idem*
Matching	1.1 ms	1.5 ms	18.3 ms	2.3 ms
Backward pass	8.2 ms	*idem*	*idem*	*idem*
Final losses	11.6 ms	11.6 ms	9.7 ms	9.7 ms

Table 5.2: Timing for each step in SSD300 on Color Boxes and a batch size of 16, computed with an Nvidia TITAN X GPU and Intel Core i7-4770K CPU @ 3.50GHz. Likewise the models we built upon, we used *Torchvision's* anchor generation implementation, which extensively relies on heavy loops and could drastically be improved (not the focus of our work). The final losses timings are partially due to the expensive hard-negative mining.

Matching	τ_2	with NMS		w/o NMS	
		AP	AR	AP	AR
Two Stage	—-	**51.6**	**67.0**	23.2	**77.8**
Unb. OT	0.01	51.1	66.3	25.3	76.5
Unb. OT	0.1	50.9	66.8	35.9	75.4
Unb. OT	1	48.3	64.4	44.3	73.4
Unb. OT	10	48.0	64.1	44.9	72.9
OT	(∞)	48.1	64.3	**45.2**	73.0

Table 5.3: Comparison of matching strategies on the Color Boxes dataset. SSD300 is evaluated both with and without NMS.

of training. Training the model for more epochs with either BM or OT does not improve metrics as the model starts to overfit. Appendix **??** provides qualitative examples and a more detailed convergence analysis. We evaluate how these results translate to faster converging DETR-like models by additionally training Deformable DETR [ZSL+21]. In addition to model improvements, Deformable DETR makes three times more predictions than DETR and uses a sigmoid focal loss [LGG+17] instead of a softmax cross-entropy loss for both classification costs. Table 5.1 gives results on Color Boxes and COCO. We observe that the entropy term does not lead to faster convergence. Indeed, Deformable DETR converges in 50 epochs with both matching strategies. Nevertheless, both OT and bipartite matching lead to similar AP and AR.

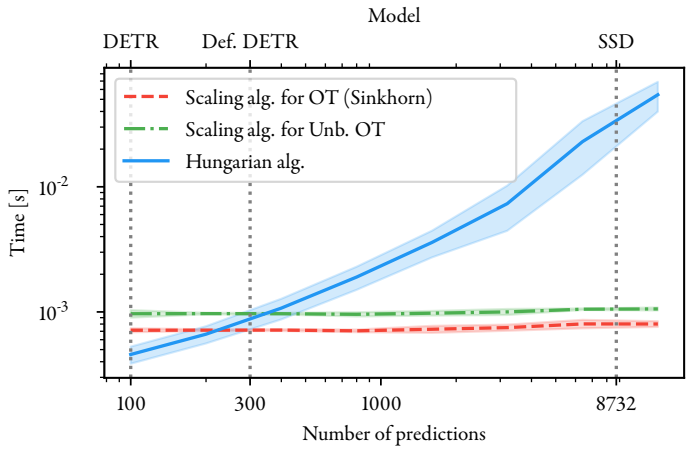


FIGURE 5.6: Average and standard deviation of the computation time for different matching strategies on COCO with batch size 16. The Hungarian algorithm is computed with *SciPy* and its time includes the transfer of the cost matrix from GPU memory to RAM. We run 20 Sinkhorn iterations. Computed with an Nvidia TITAN X GPU and Intel Core i7-4770K CPU @ 3.50GHz.

SSD and the Constraint Parameter

To better understand how unbalanced OT bridges the gap between DETR's and SSD's matching strategies, we analyze the variation in performance of SSD for different values of τ_2. Results for an initial learning rate of 0.0005 are displayed in Table 5.3. In the second row, the parameter value is close to zero. From Proposition 5.2 and when $\epsilon \to 0$, each prediction is matched to the closest ground truth box unless the matching cost exceeds 0.5. Thus, multiple predictions are matched to each ground truth box, and NMS is needed to eliminate near duplicates. When NMS is removed, AP drops by 25.8 points and AR increases by 10.2 points. We observe similar results for the original SSD matching strategy (1^{st} row), which suggests matching each ground truth box to the closest anchor box does not play a huge role in the two-stage matching procedure from SSD. The lower part of Table 5.1 shows the same for COCO. When $\tau_2 \to +\infty$, one recovers the balanced formulation used in DETR (last row). Removing NMS leads to a 2.9 points drop for AP and a 9.7 points increase for AR. Depending on the field of application, it may be preferable to apply a matching strategy with a low τ_2 and with NMS when precision is more important or without NMS when the recall is more important. Moreover, varying parameter τ_2 offers more control on the matching strategy and therefore on the precision-recall trade-off [BG94].

Computation Time

For a relatively small number of predictions, implementations of Sinkhorn perform on par with the Hungarian algorithm (Fig. 5.6). The "balanced" algorithm is on average 2.6ms slower than the Hungarian algorithm for 100 predictions (DETR) and 1.5ms faster for 300 predictions (Deformable DETR). For more predictions, GPU parallelization of the Sinkhorn algorithm makes a large difference (more than 50x speedup). As a reference point, SSD300 and SSD512 make 8,732 and 24,564 predictions.

5.5 Conclusion and Future Work

Throughout the paper, we showed both theoretically and experimentally how *Unbalanced Optimal Transport* unifies the *Hungarian algorithm*, matching each ground truth object to the best prediction and each prediction to the best ground truth, with or without threshold.

Experimentally, using OT and Unbalanced OT with entropic regularization is on par with the state-of-the-art for DETR, Deformable DETR and SSD. Moreover, we showed that entropic regularization lets DETR converge faster on the Color Boxes dataset and that parameter τ_2 offers better control of the precision-recall trade-off. Finally, we showed that the *scaling algorithms* compute large numbers of matches faster than the Hungarian algorithm.

Limitations and Future Work

The convergence improvement of the regularized OT formulation compared to bipartite matching seems to hold only for DETR and on small-scale datasets. Further investigations may include Wasserstein-based matching costs for a further unification of the theory and the reduction of the entropy with time, as it seems to boost convergence only in early phases, but not in fine-tuning.

CHAPTER 6

Recurrent Restricted Kernel Machines for Time-series Forecasting

6.1 Introduction

In [Suy17], Suykens proposed a new framework called Restricted Kernel Machines (RKM), which provides a representation of kernel methods with visible and latent variables. This representation has an objective function that is similar to the energy function of Restricted Boltzmann Machines (RBM), thus linking kernel methods with RBMs. Training and prediction requires characterizing the stationary points for the unknowns in the objective. This in turn provides the training and prediction schemes in the kernel methods setting. Restricted Kernel Machines have been previously extended for different tasks such as classification [Suy17], generation [PSS21; PFSS20] and outlier detection [TPS21]. We further extend the RKM framework to time series modeling by introducing a temporal correlation on the latent variables which provides powerful representation learning capabilities, and a novel forecasting method. The formulation draws connections with kernel autoregressive models [KHFA13] and Temporal Restricted Boltzmann Machines (TRBM) [SHT08; Oso19], which are explored in the next sections.

6.2 Recurrent Restricted Kernel Machines

6.2.1 Training

Our main objective is to capture the dynamics of a training data set $\mathcal{X}_T$ containing T time steps $\{\boldsymbol{x}_t\}_{t=1}^T \subset \mathcal{X}$. We define a feature map[1] $\phi : \mathcal{X} \to \mathcal{H}$ with $\mathcal{H}$ a (possibly infinite dimensional) Reproducing Kernel Hilbert Space (RKHS; see [SSM97] for more details). Such a feature map could be constructed explicitly or implicitly via a kernel function $k(\boldsymbol{x}, \boldsymbol{y}) : \mathcal{X}^2 \to \mathbb{R} : (\boldsymbol{x}, \boldsymbol{y}) \mapsto \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{y}) \rangle_{\mathcal{H}}$. We also define a linear operator[2] $\boldsymbol{V} : \mathbb{R}^s \to \mathcal{H}$ with $s \leq T$ and it's adjoint $\boldsymbol{V}^*$. Each datapoint $\boldsymbol{x}_t$ will be associated to a latent variable $\boldsymbol{h}_t \in \mathbb{R}^s$ through a pairing term $\langle \phi(\boldsymbol{x}_t), \boldsymbol{V}\boldsymbol{h}_t \rangle_{\mathcal{H}}$. To also capture time dependence, we only add one extra term compared to the original RKM framework [Suy17]: time correlation in the latent space using a set of non-zero lag-dependent coefficients $\mathcal{A}_T = \{a_{t,l} \,|\, 1 \leq t \leq T \text{ and } 0 \leq l < p\}$ with $p \in \mathbb{Z}^+$ a lag parameter (other coefficients are assumed to be 0). Then consider the following objective function with diagonal matrix $\boldsymbol{\Lambda}$:

$$J_T(\boldsymbol{V}, \mathcal{H}_T, \mathcal{X}_T) = \sum_{t=1}^{T} \Big[\overbrace{- \langle \phi(\boldsymbol{x}_t), \boldsymbol{V}\boldsymbol{h}_t \rangle_{\mathcal{H}}}^{\text{feature-space pairing}} \overbrace{- \sum_{l=0}^{p} a_{t,l} \boldsymbol{h}_{t-l}^\top \boldsymbol{h}_t}^{\text{temporal covariance}} + \underbrace{\frac{1}{2}\left(\boldsymbol{h}_t^\top \boldsymbol{\Lambda} \boldsymbol{h}_t + \|\phi(\boldsymbol{x}_t)\|_{\mathcal{H}}^2\right)}_{\text{regularization}} \Big] + \frac{1}{2} \mathrm{Tr}\left(\boldsymbol{V}^* \boldsymbol{V}\right). \tag{6.1}$$

Interpreting the objective function. The first two terms in (6.1) are similar to the TRBM's energy function [SHT08] which is used (along with bias terms) to define a joint-probability distribution over some visible variables $\{\boldsymbol{x} \in \mathcal{X}\}$ and latent units $\left\{\boldsymbol{h} \in \{0, 1\}^s\right\}$. It is trained with a maximum-likelihood approach where the gradients are approximated with contrastive divergence. In contrast, we propose to map the data into feature-space and center it to eliminate the need of a bias term. The first term in the objective maximizes the pairing between the visible variables in the feature-space $\left\{\phi(\boldsymbol{x}) : \boldsymbol{x} \in \mathcal{X}\right\}$ and latent variables $\{\boldsymbol{h} \in \mathbb{R}^s\}$. The second term maximizes the temporal covariance between current and past latent vectors. The regularization terms and constraints are meant to bound the objective.

[1] Throughout our discussion, we assume that the feature vectors are centered in the feature-space *i.e.* $\tilde{\phi}(\boldsymbol{x}) = \phi(\boldsymbol{x}) - \mu_\phi$ with $\mu_\phi = \mathbb{E}_{\boldsymbol{\xi} \sim \mathcal{X}}\left[\phi(\boldsymbol{\xi})\right]$. Using an implicit formulation, it suffices to notice that $\langle \tilde{\phi}(\boldsymbol{x}), \tilde{\phi}(\boldsymbol{y}) \rangle_{\mathcal{H}} = \langle \phi(\boldsymbol{x}) - \mu_\phi, \phi(\boldsymbol{y}) - \mu_\phi \rangle_{\mathcal{H}} = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{y}) \rangle_{\mathcal{H}} - \langle \mu_\phi, \phi(\boldsymbol{y}) \rangle_{\mathcal{H}} - \langle \phi(\boldsymbol{x}), \mu_\phi \rangle_{\mathcal{H}} + \langle \mu_\phi, \mu_\phi \rangle_{\mathcal{H}} = k(\boldsymbol{x}, \boldsymbol{y}) - \mathbb{E}_{\boldsymbol{\xi} \sim \mathcal{X}}\left[k(\boldsymbol{\xi}, \boldsymbol{y})\right] - \mathbb{E}_{\boldsymbol{\zeta} \sim \mathcal{X}}\left[k(\boldsymbol{x}, \boldsymbol{\zeta})\right] + \mathbb{E}_{\boldsymbol{\xi}, \boldsymbol{\zeta} \sim \mathcal{X}}\left[k(\boldsymbol{\xi}, \boldsymbol{\zeta})\right]$. In practice, we can compute statistics on $\mathcal{X}_T$.

[2] The linear operator $\boldsymbol{V}$ is often referred to as a *matrix* as it only exists explicitly in the case of finite dimensional Hilbert spaces $\mathcal{H}$. It then takes the form $\boldsymbol{V} \in \mathbb{R}^{\dim(\mathcal{H}) \times s}$ and $\boldsymbol{V}^* = \boldsymbol{V}^\top$.

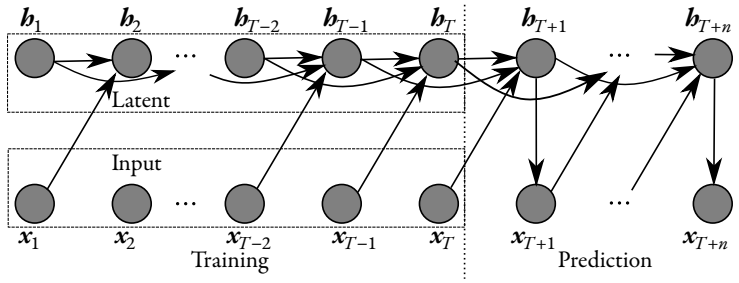


Figure 6.1: Dependency graph of the Recurrent RKM model's *training* (6.3) and *prediction* (6.8) scheme for $a_{t,l} = 1$ if $l = 1$ and $a_{t,l} = 0$ otherwise, and a linear kernel on $\mathcal{X}$.

Solving the objective. Given the visible variables, characterizing the stationary points of $J_T(\boldsymbol{V}, \mathcal{H}_T|\mathcal{X}_T)$ in the latent variables and the pairing linear operator leads to the following equations for $1 \leq t \leq T$, where $\otimes$ is the outer product:

$$\begin{cases} \dfrac{\partial J_T}{\partial \boldsymbol{V}} = -\sum_{t=1}^{T} \phi(\boldsymbol{x}_t) \otimes \boldsymbol{h}_t + \boldsymbol{V} = 0 \quad \Longrightarrow \quad \boldsymbol{V} = \sum_{t=1}^{T} \phi(\boldsymbol{x}_t) \otimes \boldsymbol{h}_t, & (6.2) \\ \dfrac{\partial J_T}{\partial \boldsymbol{h}_t} = -\boldsymbol{V}^* \phi(\boldsymbol{x}_t) + \boldsymbol{\Lambda} \boldsymbol{h}_t - \left[\sum_{l=0}^{p} a_{t,l} \boldsymbol{h}_{t-l} + \sum_{l=1}^{p} a_{t+l,l} \boldsymbol{h}_{t+l} \right] = 0. & (6.3) \end{cases}$$

Eliminating $\boldsymbol{V}$ from (6.3) using (6.2) gives the following solution

$$\left[\boldsymbol{K}(\mathcal{X}_T) + \boldsymbol{A}\right] \boldsymbol{H}^\top = \boldsymbol{H}^\top \boldsymbol{\Lambda}, \tag{6.4}$$

where $\boldsymbol{H} = \left[\boldsymbol{h}_1, \ldots, \boldsymbol{h}_T\right] \in \mathbb{R}^{s \times T}$, $\boldsymbol{A}_{i,j} = a_{i,i-j}$ for $i \geq j$ and $\boldsymbol{A}_{i,j} = a_{j-i,j}$ for $i < j$, and kernel matrix $\boldsymbol{K}(\mathcal{X}_T) = \left[k(\boldsymbol{x}_t, \boldsymbol{x}_{t'})\right]_{t,t'=1}^{T}$. We can see that any s eigenpairs of $\boldsymbol{K}(\mathcal{X}_T) + \boldsymbol{A}$ satisfies (6.4). The symmetry of $\boldsymbol{A}$ and of the kernel guarantees these eigenvalues to be real. If $\boldsymbol{A}$ is also positive semi-definite, then these eigenvalues are also guaranteed to be positive. An example of such a choice is $a_{t,l} = \exp\left(-l^2/2\sigma_t^2\right)$ for any bandwidth $\sigma_t \in \mathbb{R}^+$. Alternatively, $a_{t,l}$ can be a compactly supported function, for instance, an indicator $a_{t,l} = 1_{\{1,\ldots,p\}}(l)$ (Fig. 6.1 is an example with $p = 1$). Both these choices are however translational invariant, *i.e.* $a_{t,l} = a_l$ for any $a_{t,l} \in \mathcal{A}_T$. In other words, the local effect of time is the same at all time steps.

6.2.2 Prediction

The main idea is to generate $\{\boldsymbol{x}_{T+1}, \dots, \boldsymbol{x}_{T+n}\}$ for some $n > 0$. To do so, we now work in $\mathcal{X}_{T+n} = \mathcal{X}_T \cup \{\boldsymbol{x}_{T+1}, \dots, \boldsymbol{x}_{T+n}\}$, $\mathcal{H}_{T+n} = \mathcal{H}_T \cup \{\boldsymbol{h}_{T+1}, \dots, \boldsymbol{h}_{T+n}\}$ and consider $\mathcal{A}_{T+n}$. This gives the following objective

$$J_{T+n}(\boldsymbol{V}, \mathcal{H}_{T+n}, \mathcal{X}_{T+n}) = \sum_{t=1}^{T+n} \Big[- \langle \phi(\boldsymbol{x}_t), \boldsymbol{V}\boldsymbol{h}_t \rangle_{\mathcal{H}} - \sum_{l=0}^{p} a_{t,l} \boldsymbol{h}_{t-l}^\top \boldsymbol{h}_t + \frac{1}{2}\left(\boldsymbol{h}_t^\top \boldsymbol{\Lambda} \boldsymbol{h}_t + \|\phi(\boldsymbol{x}_t)\|_{\mathcal{H}}^2 \right) \Big] + \frac{1}{2} \mathrm{Tr}\left(\boldsymbol{V}^* \boldsymbol{V}\right). \tag{6.5}$$

Given the learned $\boldsymbol{V}$ from the training, characterizing the stationary points of $J_{T+n}\left(\mathcal{X}_{T+n}, \mathcal{H}_{T+n} \,|\boldsymbol{V}\right)$ in terms of visible and latent variables gives for $1 \leq t \leq T+n$

$$\begin{cases} \dfrac{\partial J_{T+n}}{\partial \phi\left(\boldsymbol{x}_t\right)} = -\boldsymbol{V}\boldsymbol{h}_t + \phi\left(\boldsymbol{x}_t\right) = 0 \quad \Longrightarrow \quad \phi\left(\boldsymbol{x}_t\right) = \boldsymbol{V}\boldsymbol{h}_t, & (6.6) \\ \dfrac{\partial J_{T+n}}{\partial \boldsymbol{h}_t} = -\boldsymbol{V}^* \phi(\boldsymbol{x}_t) + \boldsymbol{\Lambda}\boldsymbol{h}_t - \left[\sum_{k=0}^{p} a_{t,l} \boldsymbol{h}_{t-l} + \sum_{l=1}^{p} a_{t+l,l} \boldsymbol{h}_{t+l} \right] = 0. & (6.7) \end{cases}$$

We first notice that $\phi\left(\boldsymbol{x}_t\right) = \boldsymbol{V}\boldsymbol{h}_t$ is true for all $1 \leq t \leq T$. Furthermore, we also have $\partial J_{T+n}/\partial \boldsymbol{h}_t = \partial J_T/\partial \boldsymbol{h}_t$ for all $1 \leq t \leq T-p$. Using $\partial J_{T+n}/\partial \boldsymbol{h}_t = 0$ (6.7) and the obtained $\boldsymbol{V}$ (6.2), with $t = T-p+1$, we can find an expression for $\boldsymbol{h}_{T+1}$. Iteratively, we can find an expression for $\boldsymbol{h}_{T+m}$ with $t = T-p+m$, until $m = n$:

$$a_{T+m,t} \boldsymbol{h}_{T+m} = \left[\boldsymbol{H}\boldsymbol{A}\boldsymbol{H}^\top - a_{T+m-p,0} \mathbb{I}_s \right] \boldsymbol{h}_{T+m-p} - \Big[\sum_{l=1}^{p} a_{T+m-p,l} \boldsymbol{h}_{T+m-p-l} + \sum_{l=1}^{p-1} a_{T+m-p+l,l} \boldsymbol{h}_{T+m-p+l} \Big]. \tag{6.8}$$

This can now be used in (6.6) to find $\phi\left(\boldsymbol{x}_{T+m}\right)$, again with $t = T+m$:

$$\phi(\boldsymbol{x}_{T+m}) = \boldsymbol{V}\boldsymbol{h}_{T+m} = \sum_{t'=1}^{T} \phi(\boldsymbol{x}_{t'}) \boldsymbol{h}_{t'}^\top \boldsymbol{h}_{T+m}. \tag{6.9}$$

Finally, to obtain new data points $\{\boldsymbol{x}_t\}_{t=T+1}^{T+n}$ in the input space, the *pre-image* problem on (6.9) needs to be solved.

Solving the pre-image problem. An advantage of using a kernel function, $k(\boldsymbol{x}, \boldsymbol{y}) = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{y}) \rangle_{\mathcal{H}}$, is that all computations can be implicitly performed in feature space and

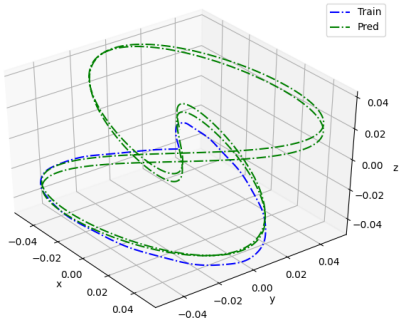


Figure 6.2: Training and predicted latent variables of a sinusoidal data set.

the exact mapping $\phi : \mathcal{X} \to \mathcal{H}$ is not required. Working with an implicit feature map however gives rise to the pre-image problem. Given a point $\boldsymbol{\psi} \in \mathcal{H}$, find $\boldsymbol{x} \in \mathcal{X}$ such that $\boldsymbol{\psi} = \phi(\boldsymbol{x})$. This pre-image problem is known to be ill-posed as the exact pre-image might not exist [SS01b]. Instead, an optimization problem is considered to find the approximate pre-image $\tilde{\boldsymbol{x}} = \arg\min_{\tilde{\boldsymbol{x}} \in \mathcal{X}} \|\boldsymbol{\psi} - \phi(\tilde{\boldsymbol{x}})\|^2_{\mathcal{H}}$. We employ two different pre-image methods in this work to solve (6.9): kernel smoother [SS18] and kernel ridge regression [WSB03].

Computational complexity. The eigendecomposition during training (see (6.4)) requires $\mathcal{O}(T^3)$ operations, the complexity of the predictions in latent space $\mathcal{O}(p)$ and in input space with the kernel-smoother $\mathcal{O}(T)$, whereas training the kernel-ridge regression $\mathcal{O}(T^3)$ since it involves solving a linear-system.

6.3 Experiments

We illustrate the representation learning capabilities by considering a simple sine wave as input to the RRKM model and exploring its latent space. Fig. 6.2 shows the latent space embedding of the learned sine wave and evolution of forecasted latent variables. Dynamics in the data are well represented in the latent space and the forecasted latent variables continue to follow the training trajectory. In Fig. 6.3, we perform an ablation study on the Santa Fe data set to identify the effect of hyper-parameters on the forecasting performance. We vary bandwidths σ_x, σ_t and latent-space dimension s. The study shows that σ_t captures phase-shift, σ_x captures amplitude and s capture higher and lower frequencies.

The proposed model is compared to a recurrent neural network (RNN) and an ARMA model which are two of the most popular methods used in time series forecasting. On each

data set [3], and method, hyperparameter tuning has been performed and the result of the best set of parameters, quantified as the mean squared error, is shown in Table 6.1. For all methods, the entire validation set is forecasted, in recursive fashion, starting from the end of the training set.

When comparing to the baseline methods, RRKM is comparable or better. The RNN can have a better result, however, due to its stochastic nature, its performance has high variability while the RRKM is deterministic for the same parameters.

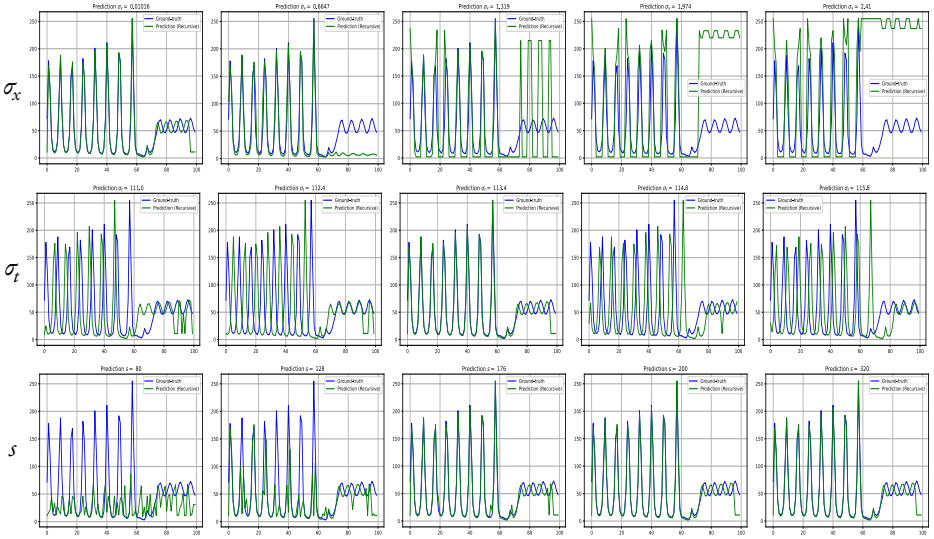


Figure 6.3: Ablation study on the Santa Fe laser data set.

Table 6.1: Mean squared error on the forecasted data. Standard deviation for 10 iterations between brackets for the stochastic models.

Data	RNN	ARMA	RRKM (Ours)
Santa Fe	3075.06 (±794.10)	2224.55	**119.06**
Chickenpox	34329.95 (±9513.07)	23571.35	**20716.91**
Energy	16002.11 (±1809.89)	24797.40	**12764.097**
Turbine	2401.12 (±644.53)	1317.67	**1299.915**

[3] `https://rdrr.io/cran/TSPred/man/SantaFe.A.html`, `https://archive.ics.uci.edu/ml/datasets/Hungarian+Chickenpox+Cases`, `https://archive.ics.uci.edu/ml/datasets/Appliances+energy+prediction`, `https://archive.ics.uci.edu/ml/datasets/Gas+Turbine+CO+and+NOx+Emission+Data+Set`.

6.4 Conclusion

In this work, we introduced the recurrent restricted kernel machine model. This framework provides new insights and ideas for time series modeling including latent space dynamics and a novel forecasting method. For future work, we believe a further exploration of the representation learning capabilities in latent space can provide new ways to interpret the data. Additionally, besides the topics mentioned in this work, the framework can be extended towards other tasks involving time series such as denoising, handling missing values and classification.

CHAPTER 7

Conclusion

APPENDIX A

Overview of the Datasets Used

A.1 USPS Handwritten Digits

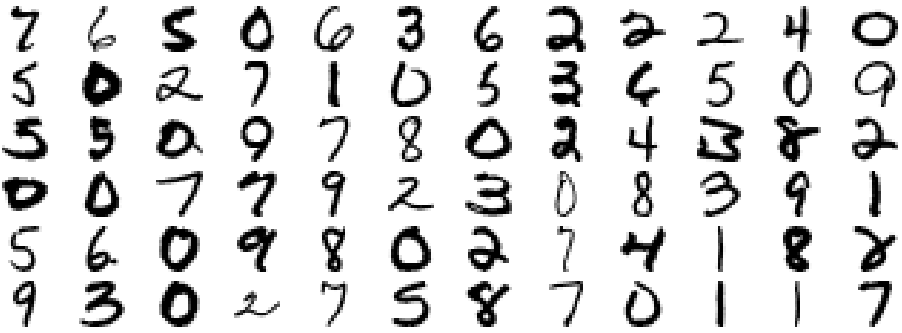


FIGURE A.1: Sample images from the USPS Handrwitten Digits dataset.

A.2 MNIST

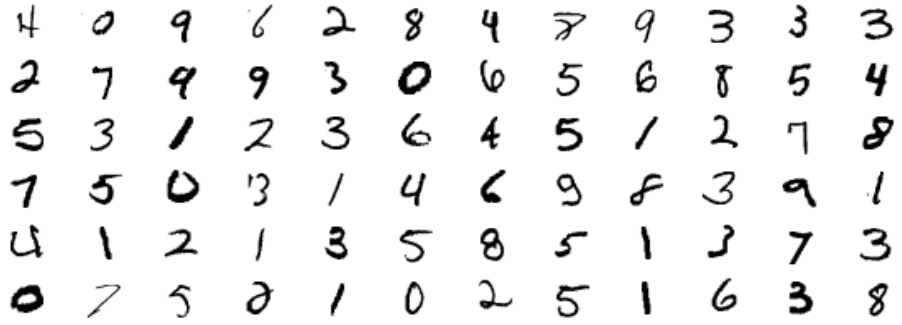


Figure A.2: Sample images from the MNIST dataset.

A.3 Quickdraw

Figure A.3: Sample images from the Quickdraw dataset.

A.4 COCO

Short name for *Common Objects in Context*.

A.5 Color Boxes

This section provides a discussion of the Color Boxes synthetic dataset. It is split into 4,800 training and 960 validation images of 500 × 400 pixels. Images have a gray background. We

uniformly randomly draw between 0 and 30 rectangles of 20 different colors, which define the category of the rectangle. The dimension of the rectangles vary from 12 to 80 pixels and are uniformly randomly rotated. They are placed such that the IoU between their bounding boxes is at most 0.25. A gaussian noise of mean 0 and standard deviation 0.05 is added to each pixel value independently. Sample images are drawn in Fig. A.4.

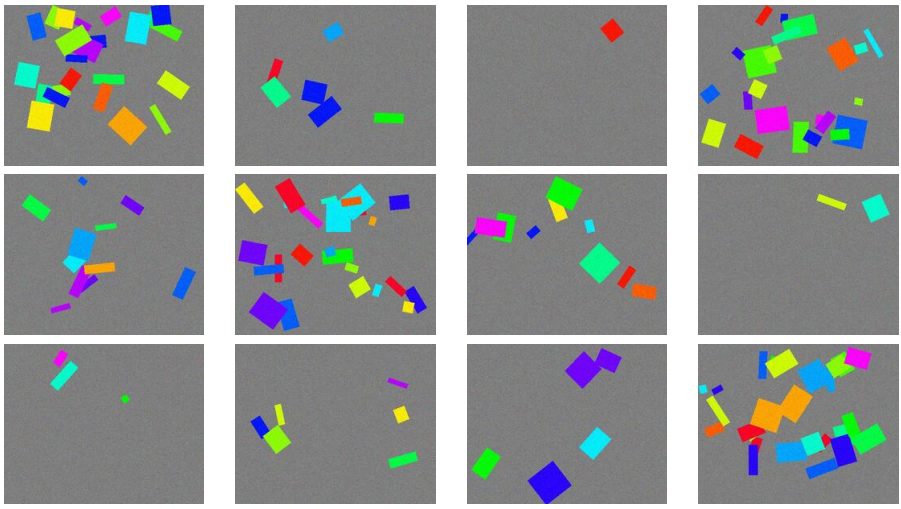


Figure A.4: Sample images from the Color Boxes dataset.

Bibliography

[ABGR19] M. Z. Alaya, M. Berar, G. Gasso, and A. Rakotomamonjy, "Screening sinkhorn algorithm for regularized optimal transport", in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[ACB17] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks", in *International conference on machine learning*, PMLR, 2017, pp. 214–223.

[ANR17a] J. Altschuler, J. Niles-Weed, and P. Rigollet, "Near-linear time approximation algorithms for optimal transport via sinkhorn iteration", in *Advances in Neural Information Processing Systems 30*, Curran Associates, Inc., 2017, pp. 1964–1974.

[ANR17b] J. Altschuler, J. Niles-Weed, and P. Rigollet, "Near-linear time approximation algorithms for optimal transport via sinkhorn iteration", in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30, Curran Associates, Inc., 2017.

[BCR84] C. Berg, J. P. R. Christensen, and P. Ressel, *Harmonic Analysis on Semigroups* (Graduate Texts in Mathematics). New York, NY: Springer New York, 1984, vol. 100.

[BG94] M. Buckland and F. Gey, "The relationship between recall and precision", *Journal of the American society for information science*, vol. 45, no. 1, pp. 12–19, 1994.

[CCO17] M. Carrière, M. Cuturi, and S. Oudot, "Sliced wasserstein kernel for persistence diagrams", in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ser. ICML'17, Sydney, NSW, Australia: JMLR.org, 2017, pp. 664–673.

[Chi17] L. Chizat, "Unbalanced Optimal Transport : Models, Numerical Methods, Applications", Theses, Université Paris sciences et lettres, Nov. 2017.

[CJM19] D. Chen, L. Jacob, and J. Mairal, "Biological sequence modeling with convolutional kernel networks", *Bioinformatics*, vol. 35, no. 18, pp. 3294–3302, Feb. 2019.

[CMS+20] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers", in *European conference on computer vision*, Springer, 2020, pp. 213–229.

[CPSV18a] L. Chizat, G. Peyré, B. Schmitzer, and F.-X. Vialard, "Scaling algorithms for unbalanced optimal transport problems", *Mathematics of Computation*, vol. 87, no. 314, pp. 2563–2609, 2018.

[CPSV18b] L. Chizat, G. Peyré, B. Schmitzer, and F.-X. Vialard, "Unbalanced optimal transport: Dynamic and kantorovich formulations", *Journal of Functional Analysis*, vol. 274, no. 11, pp. 3090–3123, 2018.

[Cut13a] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport", in *Advances in Neural Information Processing Systems 26*, Curran Associates, Inc., 2013, pp. 2292–2300.

[Cut13b] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport", *Advances in neural information processing systems*, vol. 26, 2013.

[CV18] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6154–6162.

[DFS20] H. De Plaen, M. Fanuel, and J. A. Suykens, "Wasserstein exponential kernels", in *2020 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2020, pp. 1–6.

[DLHS16] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks", *Advances in neural information processing systems*, vol. 29, 2016.

[DN16] K. Date and R. Nagi, "Gpu-accelerated hungarian algorithms for the linear assignment problem", *Parallel Computing*, vol. 57, pp. 52–72, 2016.

[EK72] J. Edmonds and R. M. Karp, "Theoretical improvements in algorithmic efficiency for network flow problems", *J. ACM*, vol. 19, no. 2, pp. 248–264, Apr. 1972.

[FB12] B. O. Fagginger Auer and R. H. Bisseling, "A gpu algorithm for greedy graph matching", in *Facing the Multicore-Challenge II*, Springer, 2012, pp. 108–119.

[FSFC21] K. Fatras, T. Séjourné, R. Flamary, and N. Courty, "Unbalanced minibatch optimal transport; applications to domain adaptation", in *International Conference on Machine Learning*, PMLR, 2021, pp. 3186–3197.

[FZM+15] C. Frogner, C. Zhang, H. Mobahi, M. Araya, and T. A. Poggio, "Learning with a wasserstein loss", *Advances in neural information processing systems*, vol. 28, 2015.

[GAA+17] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans", *Advances in neural information processing systems*, vol. 30, 2017.

[GDDM14] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[Gir15] R. Girshick, "Fast r-cnn", in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Dec. 2015, pp. 1440–1448.

[GLL+21] Z. Ge, S. Liu, Z. Li, O. Yoshie, and J. Sun, "Ota: Optimal transport assignment for object detection", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 303–312.

[GLW+21] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "Yolox: Exceeding yolo series in 2021", *arXiv preprint arXiv:2107.08430*, 2021.

[HGDG17] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn", in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[HLS+20] Y. Han, X. Liu, Z. Sheng, Y. Ren, X. Han, J. You, R. Liu, and Z. Luo, "Wasserstein loss-based deep object detection", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Jun. 2020.

[HMHS17] X. Huang, A. Maier, J. Hornegger, and J. A. K. Suykens, "Indefinite kernels in least squares support vector machines and principal component analysis", *Applied and Computational Harmonic Analysis*, vol. 43, no. 1, pp. 162–172, 2017.

[Hul94] J. J. Hull, "A database for handwritten text recognition research", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 5, pp. 550–554, May 1994.

[HZRS15] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition", *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[Kan58] L. Kantorovitch, "On the translocation of masses", *Management Science*, vol. 5, no. 1, pp. 1–4, 1958.

[KHFA13] M. Kallas, P. Honeine, C. Francis, and H. Amoud, "Kernel autoregressive models using Yule-Walker equations", *Signal Processing*, vol. 93, no. 11, pp. 3053–3061, 2013.

[KNS+19] S. Kolouri, K. Nadjahi, U. Simsekli, R. Badeau, and G. Rohde, "Generalized sliced wasserstein distances", in *Advances in Neural Information Processing Systems 32*, Curran Associates, Inc., 2019, pp. 261–272.

[KPMR18] S. Kolouri, P. E. Pope, C. E. Martin, and G. K. Rohde, "Sliced wasserstein auto-encoders", in *International Conference on Learning Representations*, 2018.

[Kuh55] H. W. Kuhn, "The hungarian method for the assignment problem", *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[KZR16] S. Kolouri, Y. Zou, and G. K. Rohde, "Sliced wasserstein kernels for probability distributions", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5258–5267.

[LAE+16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector", in *European conference on computer vision*, Springer, 2016, pp. 21–37.

[LBR20] J. Lee, N. P. Bertrand, and C. J. Rozell, "Unbalanced optimal transport regularization for imaging problems", *IEEE Transactions on Computational Imaging*, vol. 6, pp. 1219–1232, 2020.

[LC10] Y. LeCun and C. Cortes, "MNIST handwritten digit database", 2010.

[LDG+17] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[LGG+17] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection", in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[LLZ+21] S. Liu, F. Li, H. Zhang, X. Yang, X. Qi, H. Su, J. Zhu, and L. Zhang, "Dab-detr: Dynamic anchor boxes are better queries for detr", in *International Conference on Learning Representations*, 2021.

[LMB+14] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context", in *European conference on computer vision*, Springer, 2014, pp. 740–755.

[LOW+20] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey", *International journal of computer vision*, vol. 128, no. 2, pp. 261–318, 2020.

[LZL+22] F. Li, H. Zhang, S. Liu, J. Guo, L. M. Ni, and L. Zhang, "Dn-detr: Accelerate detr training by introducing query denoising", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 13 619–13 627.

[Mai16] J. Mairal, "End-to-end kernel learning with supervised convolutional kernel networks", in *Advances in Neural Information Processing Systems 29*, Curran Associates, Inc., 2016, pp. 1399–1407.

[MMC16] G. Montavon, K.-R. Müller, and M. Cuturi, "Wasserstein training of restricted boltzmann machines", *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[Mon81] G. Monge, "Mémoire sur la théorie des déblais et des remblais", *Mem. Math. Phys. Acad. Royale Sci.*, pp. 666–704, 1781.

[Mun57] J. Munkres, "Algorithms for the assignment and transportation problems", *Journal of the society for industrial and applied mathematics*, vol. 5, no. 1, pp. 32–38, 1957.

[OMCS04] C. S. Ong, X. Mary, S. Canu, and A. J. Smola, "Learning with non-positive kernels", in *Twenty-first international conference on Machine learning - ICML '04*, New York, New York, USA: ACM Press, 2004, p. 81.

[Oso19] T. Osogami, "Boltzmann machines for time-series", *CoRR*, vol. abs/1708.06004, 2019.

[OTN+22] M. Otani, R. Togashi, Y. Nakashima, E. Rahtu, J. Heikkilä, and S. Satoh, "Optimal correction cost for object detection evaluation", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 21 107–21 115.

[PC+19] G. Peyré, M. Cuturi, *et al.*, "Computational optimal transport: With applications to data science", *Foundations and Trends® in Machine Learning*, vol. 11, no. 5-6, pp. 355–607, 2019.

[PC19] G. Peyré and M. Cuturi, *Computational Optimal Transport: With Applications to Data Science*. Now Foundations and Trends, 2019.

[PCS+19] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, "Libra r-cnn: Towards balanced learning for object detection", in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 821–830.

[PFSS20] A. Pandey, M. Fanuel, J. Schreurs, and J. A. K. Suykens, "Disentangled representation learning and generation with manifold optimization", *To appear in Neural Computation. CoRR*, vol. abs/2006.07046, 2020.

[PSS] A. Pandey, J. Schreurs, and J. A. K. Suykens, *Generative restricted kernel machines*.

[PSS21] A. Pandey, J. Schreurs, and J. A. Suykens, "Generative restricted kernel machines: A framework for multi-view generation and disentangled feature learning", *Neural Networks*, vol. 135, pp. 177–191, 2021.

[RDGF16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[RHGS15] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks", in *Advances in Neural Information Processing Systems*, vol. 28, Curran Associates, Inc., 2015.

[RST18] P. K. Rubenstein, B. Schoelkopf, and I. Tolstikhin, "On the latent space of wasserstein auto-encoders", *arXiv preprint arXiv:1802.03761*, 2018.

[RTG+19] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression", in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA: IEEE Computer Society, Jun. 2019, pp. 658–666.

[RW06] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[San15] F. Santambrogio, *Optimal Transport for Applied Mathematicians: Calculus of Variations, PDEs, and Modeling* (Progress in Nonlinear Differential Equations and Their Applications). Springer International Publishing, 2015.

[Sch19] B. Schmitzer, "Stabilized sparse scaling algorithms for entropy regularized transport problems", *SIAM Journal on Scientific Computing*, vol. 41, no. 3, A1443–A1481, 2019.

[SGB+02] J. A. K. Suykens, T. V. Gestel, J. D. Brabanter, B. D. Moor, and J. Vandewalle, *Least Squares Support Vector Machines*. Singapore: World Scientific, 2002.

[SHT08] I. Sutskever, G. E. Hinton, and G. W. Taylor, "The recurrent temporal restricted boltzmann machine", in *Advances in Neural Information Processing Systems*, vol. 21, 2008.

[SS01a] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2001.

[SS01b] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2001.

[SS18] J. Schreurs and J. A. K. Suykens, "Generative kernel PCA", in *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2018, pp. 129–134.

[SSM97] B. Schölkopf, A. Smola, and K.-R. Müller, "Kernel principal component analysis", in *International conference on artificial neural networks*, Springer, 1997, pp. 583–588.

[Suy17] J. A. K. Suykens, "Deep restricted kernel machines using conjugate feature duality", en, *Neural Computation*, vol. 29, no. 8, pp. 2123–2163, 2017.

[SV] M. Snow and J. Van Lent, *Monge's optimal transport distance for image classification*.

[SV99a] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers", *Neural Process. Lett.*, vol. 9, no. 3, pp. 293–300, Jun. 1999.

[SV99b] J. A. K. Suykens and J. Vandewalle, "Multiclass least squares support vector machines", in *IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No.99CH36339)*, vol. 2, Jul. 1999, 900–903 vol.2.

[TBGS18] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schoelkopf, "Wasserstein auto-encoders", in *International Conference on Learning Representations*, 2018.

[TPS21] F. Tonin, P. Patrinos, and J. A. Suykens, "Unsupervised learning of disentangled representations in deep restricted kernel machines with orthogonality constraints", *Neural Networks*, vol. 142, pp. 661–679, 2021.

[Vil08] C. Villani, *Optimal Transport: Old and New* (Grundlehren der mathematischen Wissenschaften). Springer Berlin Heidelberg, 2008.

[Vil09] C. Villani, *Optimal transport: old and new*. Springer, 2009, vol. 338.

[VJ22] X.-T. Vo and K.-H. Jo, "A review on anchor assignment and sampling heuristics in deep learning-based object detection", *Neurocomputing*, 2022.

[VR09] C. N. Vasconcelos and B. Rosenhahn, "Bipartite graph matching computation on gpu", in *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, Springer, 2009, pp. 42–55.

[WSB03] J. Weston, B. Schölkopf, and G. Bakir, "Learning to find pre-images", in *Advances in Neural Information Processing Systems*, vol. 16, 2003.

[YYM+21] X. Yang, J. Yan, Q. Ming, W. Wang, X. Zhang, and Q. Tian, "Rethinking rotated object detection with gaussian wasserstein distance loss", in *Proceedings of the 38th International Conference on Machine Learning*, M. Meila and T. Zhang, Eds., ser. Proceedings of Machine Learning Research, vol. 139, PMLR, 18–24 Jul 2021, pp. 11 830–11 841.

[ZSL+21] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection", in *International Conference on Learning Representations*, 2021.

List of publications

Input file chapters/publications/publications.tex does not exist. Make sure its starts with "\chapter{List of publications}". To not include this chapter in the table of contents, use the starred version of the \chapter command...

FACULTY OF ENGINEERING SCIENCE
DEPARTMENT OF ELECTRICAL ENGINEERING (ESAT)
STADIUS CENTER FOR DYNAMICAL SYSTEMS, SIGNAL PROCESSING AND DATA ANALYTICS
Kasteelpark Arenberg 10 box 2446
B-3001 Leuven
https://www.esat.kuleuven.be/stadius/